NOLTR 62-168
DASA-1518

ONE-DIMENSIONAL HYDRODYNAMIC CODE FOR NUCLEAR-EXPLOSION CALCULATIONS

MARCH 1965

NOL

UNITED STATES NAVAL ORDNANCE LABORATORY, WHITE OAK, MARYLAND

NOLTR 62-168

NOLTR 62-168   DASA-1518

ONE-DIMENSIONAL HYDRODYNAMIC CODE FOR NUCLEAR-EXPLOSION CALCULATIONS

Prepared by:
D. Lehto
M. Lutzky

ABSTRACT: A one-dimensional hydrodynamic finite-difference computer code (WUNDY), which evolved from the M. Wilkins KO-CODE of the University of California Radiation Laboratory, is described and applied to the calculation of the shockwave from a nuclear explosion. FORTRAN IV listings are given.

PUBLISHED MARCH 1965

Air-Ground Explosions Division
Explosions Research Department
U. S. NAVAL ORDNANCE LABORATORY
WHITE OAK, SILVER SPRING, MARYLAND

ONE-DIMENSIONAL HYDRODYNAMIC CODE FOR NUCLEAR-EXPLOSION CALCULATIONS

A number of requests have been received by this Laboratory for complete details of the one-dimensional hydrodynamic code used for the nuclear-explosion calculations of references 3 and 4. This report has been prepared to satisfy these requests. The FORTRAN IV listings given in the appendices should be sufficient to enable the reader to duplicate this code, if he so desires.

This version of the WUNDY code has been written explicitly for publication; it is not the version actually used for references 3 and 4. A number of errors may have escaped detection of the authors in the print-outs of this report. The authors welcome notification of any errors found.

Support for development of the WUNDY (ref. 1) code for nuclear explosions was provided by the Defense Atomic Support Agency through Nuclear Weapons Research Subtask 01.003 (Task NOL-181).

R. E. ODENING
Captain, USN
Commander

C. J. ARONSON
By direction

CONTENTS

1. INTRODUCTION

This report describes the one-dimensional hydrodynamic code, WUNDY, that is being used at the Naval Ordnance Laboratory for nuclear-explosion calculations.

The first FORTRAN version of WUNDY, written by W. Walker (ref. 1), was based on the KO-CODE of the University of California Radiation Laboratory (ref. 2). Several versions of WUNDY now exist at this Laboratory. The one described in this report has been used in the calculation of some of the hydrodynamic aspects of nuclear explosions in air (ref. 3, 4).

2. DESCRIPTION OF THE WUNDY CODE

2.1 General Properties: A one-dimensional hydrodynamic code is basically quite simple. It is the refinements that make it seem lengthy. Many options are usually included; for example, plane, cylindrical and spherical symmetries are usually included in the same code. Features like a variable, internally calculated time step are desirable. Special computations are often included, such as the x-ray deposition of PUFF (refs. 5, 6, 7, 8). Equations of state can become quite long. The input and output routines usually occupy a good deal of space.

WUNDY as given here is in cgs units. It handles only pure hydrodynamics (no radiative transfer, conduction, etc.) in plane, cylindrical or spherical geometry. The geometry of the code is collected into a single accessible place where it can be modified for calculations requiring area to be some function of radius other than $A = cr^n$ with $n = 0$, 1, or 2. Shocks are handled by the artificial viscosity method. WUNDY has 300 zones, divided among as many as 30 different regions or materials. The version described in this report is written for clarity rather than for maximum computing efficiency. For example, constants of proportionality (e.g., pi) are retained even though they would eventually cancel out. Thus, in spherical geometry, the "mass" of a zone is always the actual number of grams of material in that zone.

A model of the earth's atmosphere (ref. 9) is included as a subroutine and can be used to furnish initial ambient atmospheric conditions. Calculations can be performed for shockwave propagation in an arbitrary direction from an explosion at an arbitrary altitude of burst (e.g., the shockwave propagation down to the ground from a high altitude explosion can be calculated). The component of the earth's gravity along the calculated ray is included in the equations of motion.

2.2 The Finite Difference Equations: The problem is advanced from time t^n to time t^{n+1} by the usual finite difference equations (refs. 10, 11, 12, 13):

$$u_j^{n+1/2} = u_j^n + \Delta t_j^n \frac{(P_{j-1/2}^n - P_{j+1/2}^n)}{1/2(m_{j-1/2} + m_{j+1/2})} A_j^n - g \qquad (1)$$

$$x_j^{n+1} = x_j^n + \Delta t_j^{n+1/2} u_j^{n+1/2} \qquad (2)$$

$$v_{j-1/2}^{n+1} = V_{j-1/2}^{n+1} / m_{j-1/2} \qquad (3)$$

$$p_{j-1/2}^{n+1} = \frac{2e_{j-1/2}^n - (p_{j-1/2}^n + 2\, q_{j-1/2})(v_{j-1/2}^{n+1} - v_{j-1/2}^n)}{\frac{\gamma+1}{\gamma-1} v_{j-1/2}^{n+1} - v_{j-1/2}^n} \qquad (4)$$

$$e_{j-1/2}^{n+1} = \frac{p_{j-1/2}^{n+1} v_{j-1/2}^{n+1}}{(\gamma-1)} \qquad (5)$$

where
- x = distance from origin
- u = velocity of interface
- v = specific volume of zone
- p = pressure in zone
- q = artificial viscosity in zone
- $P = p + q$
- e = internal energy in zone
- γ = equation of state constant = 1.4 for ideal air
- g = acceleration of gravity component

A_j^n = area of interface j at time n, $= 4\pi(x_j^n)^2$ for sphere

$V_{j-1/2}^n$ = volume of zone $j-1/2 = \frac{4}{3}\pi\left[(x_j^n)^3 - (x_{j-1}^n)^3\right]$ for sphere

Δt_j^n = timestep = $1/2\ (\Delta t_j^{n+1/2} + \Delta t_j^{n-1/2})$

m = mass of zone

These equations are discussed in the references; this material will not be repeated here. The equations for p and e are for a gamma-law gas, $e = pv/(\gamma-1)$.

2.3 Initialization: A problem is initialized when each interface has been assigned a position and velocity, each zone between interfaces has been assigned a mass and internal energy, and all of the control variables have been specified. Most problems are completely initialized by a deck of about 18 cards. The first two cards contain lettering to be printed on the output. The next four cards contain the control variables (printing frequency, type of geometry, etc.). These are followed by a group of cards (one for eadh region) describing each region of the problem (a region is usually a group of zones of the same material). The dimensions, velocities, densities, number of zones in each region, etc. are given on these cards. The normal input deck ends with ten cards giving some additional data (direction of calculated ray, altitude of burst, etc.).

In some cases one wishes to give initial data for interfaces or zones individually rather by regions. For example, in a moving-piston problem, the piston would be introduced by giving the first interface a velocity. This is done by additional cards following the normal input deck.

Additional input can also be read from tape. Special input calculations (e.g., in high explosive problems, the initial conditions in the detonation wave) can be made by other programs and written on tape that WUNDY can read. WUNDY can also write its own tape from any record of which the problem can be resumed later.

2.4 Rezoning: The computer time required for a problem is about proportional to the number of zones that are being calculated and about inversely proportional to the width of the smallest zone in the problem. Four types of rezoning have been used in reducing computing times. A general rezoning routine that contains all four of these methods as special cases could probably be devised.

The first type of rezoning becomes alerted when the timestep begins to decrease appreciably. In some problems this decrease is due to excessive compression of a zone that does not lie in the region of interest. This rezoning method (REZ1) seeks out the zone causing the small timestep and removes it by combining it with an adjacent zone.

A second type of rezoning (REZ2) waits until a disturbance (usually the main shockwave) reaches the outer boundary of the problem and then moves this boundary out by a factor of 2 in distance. It then combines pairs of adjacent zones to release enough zones to fill the new extra region of space out to the new boundary. This allows the same number of zones to cover an indefinitely expanding space, at the cost of resolution lost by making the zones larger. This is the type of rezoning used in the calculations of references 3 and 4.

A third type of rezoning (REZ3) keeps down the total number of zones required for a problem by having fairly large zones everywhere except in the region of interest (usually the main shockwave). The shockwave is kept supplied with fine zones by subdividing large ambient zones as the shock approaches and then recombining them into large zones when the shock has passed. This scheme is now in use on nearly every problem.

A fourth type of rezoning (REZ4) replaces the zones of the first (fireball) region by a smaller number of zones. In some problems the first region holds down the timestep needlessly long after the shockwave has moved away. In these cases the problem can be made to run faster by a factor of two or three by replacing the spent fireball by just a few large zones.

2.5 Summary Routines: Two subroutines have been included to provide data in addition to the normal output. Subroutine OUT2 prints tables of the shock location, pressure, etc. as a function of time. This output is useful in preparing shock front pressure versus distance curves and shock front radius versus time curves. Subroutine OUT3 prints tables of the pressure, density, particle velocity, dynamic pressure, etc. versus time at fixed distances (compared to the normal output, which gives quantities versus distance at fixed times). This output is useful for predicting the pressure-time records, etc. at fixed gage locations. An option is included for collecting data at points moving in straight-line trajectories near the explosion.

2.6 Equations of State: Two equations of state are included in the listings of Appendix B. One (EQS1) is for an ideal gas of constant gamma. The other (EQS2) is an approximate equation of state for real air, based on reference 14. These can be used as patterns for preparing equations of state for other materials. The essential things to be calculated are the pressure, internal energy, and an approximate sound speed (needed in calculating the timestep). The temperature and gamma may be calculated if desired but are not needed elsewhere in the program.

3. USING THE CODE

3.1 Initial Conditions for Nuclear Explosion Problems: One approximation to a nuclear explosion that has been found useful (refs. 3 and 4) is a uniform sphere of highly heated air that is allowed to expand hydrodynamically into the surrounding ambient air.

The energy of the explosion (adjusted, if desired, for early losses by thermal radiation) is assumed to be uniformly distributed within a sphere whose radius in meters can be taken as roughly

$$R_o = 4.251 \left(\frac{W}{p_o}\right)^{1/3} , \tag{6}$$

where W is the energy within the sphere in kilotons and p_o is the ambient pressure in atmospheres. R_o is roughly the radius at which radiative expansion gives way to a shockfront at the periphery of the fireball. The properties of the main shockwave are not strongly dependent on the choice of initial sphere size (ref. 4). Deviating from the above value by even a factor of 5 does not affect the main shock seriously, provided that one does not use the computed shock-wave values until the initial sphere has about doubled its radius. Some time is needed for the energy to be fed into the shockwave.

The weapon mass is usually neglected in these calculations. It is usually fairly small compared to the mass of air within the initial sphere. Actual details of the weapon expansion can be included if the initial configurations and equations of state for the weapon materials are incorporated.

The energy to put into the initial hot-air sphere is just

$$E(\text{erg/gram}) = 4.185 \times 10^{19}\, W \frac{1}{\frac{4}{3}\pi R_o^3 \rho} + E_o \tag{7}$$

where W is again the energy in kilotons and ρ is the air density within the fireball (usually assumed to equal ambient density). E_o is the energy of the ambient air before the explosion; it is usually negligible.

The continuous actual problem must be divided into zones for the finite difference solution. If these zones are made small, the problem will run slowly. If the zones are made larger to make the problem run faster, the results will be less accurate. When the zones are too large, the shock tends to have too high a pressure and to travel too fast. A useful compromise between computing time and accuracy has been found to be a zone size, in the ambient air ahead of the shock, of roughly

$$\Delta x = 1.0\ (W/p_o)^{1/3} \text{ meters.} \tag{8}$$

3.2 Other Problems: This code is, of course, not restricted to nuclear explosion calculations. The main adaptations for nuclear explosion use are the ARDC atmosphere, the rezoning routines, and the summary routines. Given appropriate equations of state, the code can be used for a variety of one-dimensional problems, e.g., high explosives in vacuum, air, water, or any medium whose equation of state is known.

4. REFERENCES

(1) Walker, W. A., Piacesi, D., and Sternberg, H. M., "A Fortran Program for the Numerical Solution of Explosion and Shock Hydrodynamic Problems in One Space Dimension," U. S. Naval Ordnance Laboratory NOLTR 64-114, to be published. Unclassified.

(2) Wilkins, M., et al, "A Computer Program for Calculating One-Dimensional Hydrodynamic Flow-KO CODE," University of California Radiation Laboratory, UCRL 6919, July 1962. Unclassified.

(3) Lutzky, M., Lehto, D., and Rudlin, L., "Hydrodynamic Calculations of the Shockwave from Nuclear Explosions at Sea-Level Altitude of Burst," U. S. Naval Ordnance Laboratory, NOLTR 62-152, 14 Sep 1962. Unclassified.

(4) Lutzky, M., Lehto, D., and Rudlin, L., "Hydrodynamic Calculations of the Shockwave from Nuclear Explosions at 20-km Altitude of Burst (U)," U. S. Naval Ordnance Laboratory, NOLTR 62-153, 17 Sep 1962. Secret Restricted Data.

(5) Welch, J. A., et al, "Notes on the AFSWC Hydrocode, "PUFF" (U)," Air Force Special Weapons Center, SWRPT-1-0076, 11 Oct 61. Confidential.

(6) Swarstrauber, P., et al, "Notes on the AFSWC Hydrocode, "PUFF", Supplement 1 (U)," Air Force Special Weapons Center, SWRPT-1-0079, 11 Oct 61. Confidential.

(7) "Listings of the AFSWC Hydrocode, "PUFF" (U)," Air Force Special Weapons Center, SWRPT-1-0078, 11 Oct 61. Confidential.

(8) Grandey, R. A., "PUFF-VTS Computer Program (U)," Air Force Special Weapons Center, AFSWC-TDR-62-76, Feb 63. Confidential.

(9) Minzner, R. A., and Ripley, W. S., "The ARDC Model Atmosphere," Geophysics Research Directorate, Air Force Cambridge Research Center, AFCRC TN-56-204, 1956. Unclassified.

(10) Richtmyer, R. D., Difference Methods for Initial-Value Problems, Interscience, New York, 1957.

(11) Fox, L., Numerical Solution of Ordinary and Partial Differential Equations, Pergamon Press, Ltd., Oxford, 1962.

(12) Forsythe, G. E. and Wasow, W. R., Finite-Difference Methods for Partial Differential Equations, Wiley, New York, 1960.

(13) Piacesi, R., et al, "A Computer Analysis of the Two-Stage Hypervelocity Model Launchers," U. S. Naval Ordnance Laboratory, NOLTR 62-87, 20 Feb 63. Unclassified.

(14) Hilsenrath, J., et al, "Thermodynamic Properties of Highly Ionized Air," Air Force Special Weapons Center, AFSWC-TR-56-35, ASTIA AD-96303, Apr 57. Unclassified.

Copy to:
J. F. Ablard
Cdr. Oleksy
C. J. Aronson
F. Swift, Jr.
I. Kabik
J. F. Moulton, Jr.
D. L. Lehto
M. Lutzky
L. Rudlin
H. Sternberg
W. Walker
D. Piacesi
R. Piacesi
M. St. Jean
J. F. Goertner
J. W. Enig
F. J. Petrone
C. F. Dieter
H. Hurwitz
R. C. Roberts
H. G. Snay

APPENDIX A

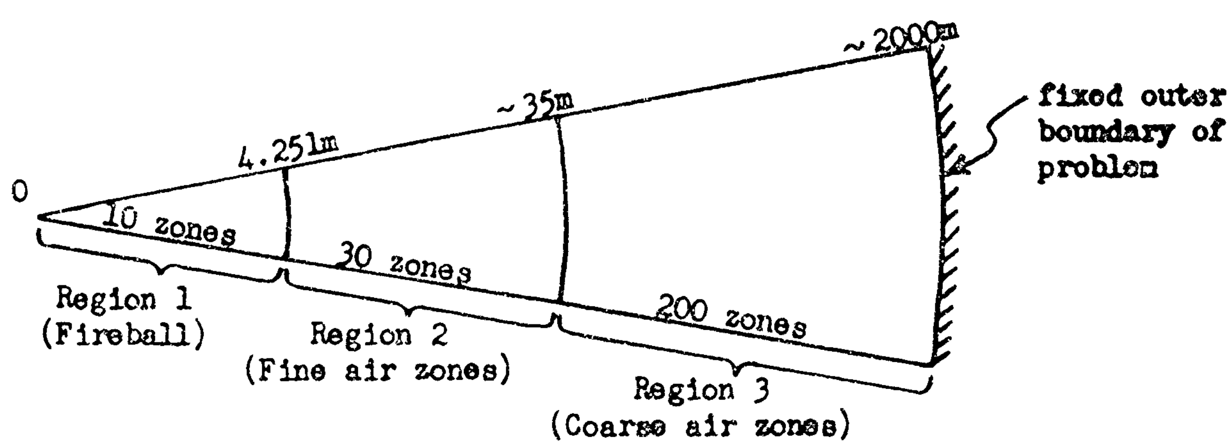
Example: 1 Kt Explosion at Sea Level

The problem must be divided into zones and regions and initial values must be assigned to x, u, e, and density (1/v) for each zone. It is convenient to divide this problem into three uniform regions as shown in the figure below: the fireball, a region of fine zones for the shockwave to move into, and an outermost extensive region of coarse zones which will be subdivided into fine zones as the shockwave moves out.

Equation (6) gives a fireball radius of $R_0 = 4.251$ meters. Equation (8) gives one meter as a suitable zone size in the fine zone region. Let there be ten fine zones per coarse zone and use 30 fine zones. Then region 2 is 30 meters thick and the radius of its outer boundary is 34.251 meters. The large zones in the third region are 10 meters thick. Using 200 of them would give a problem boundary of 2034.251 meters, which is far enough to contain the phenomena of interest. These numbers need not be so exact; 35 meters and 2000 meters are close enough.

The initial internal energy in the fireball, from equation (7), is 1.062×10^{14} erg/gram. The internal energy of the ambient air, calculated from $e = pv/(\gamma - 1)$, is 2.068×10^{9} erg/gram.

A sector of this problem appears as follows:



THE DATA FOR THIS PROBLEM MIGHT BE AS FOLLOWS-

```
CARD 1            WUNDY CODE CALCULATION FOR 1 KT AT SEA LEVEL.

CARD 2            U.S.N.O.L.   3-1-65    1 KT AT SEA LEVEL.

CARD 3  NPROB      1160  (ARBITRARY PROBLEM NUMBER)
        K             3  (SPHERICAL GEOMETRY)
        MURIN         2  (ORIGIN IS FIXED)
        MUREX         2  (OUTER BOUNDARY OF PROBLEM IS FIXED)
        IMAX          3  (THREE REGIONS)
        IARDC         0  (ARDC ATMOSPHERE NOT USED)
        INTAPE        0  (NO DATA INPUT FROM TAPE 18)
        INCODS        0  (NO EXTRA DATA INPUT FROM CARDS)
        NQUIT      4000  (STOP PROBLEM AT CYCLE 4000)
        NPR          50  (PRINT AFTER EVERY 50 CYCLES)
        NTAPE         0  (DO NOT WRITE TAPE 18)
        JCALC        20  (START WITH ONLY THE FIRST 20 ZONES ACTIVE)

CARD 4  KOUT2         4  (USE 8 POINTS TO GET EXTRAPOLATED PRESSURE)
        KOUT2A       10  (STORE SHOCK FRONT DATA EVERY 10 CYCLES)
        KOUT2B        0  (NO INTERACTION BETWEEN OUT2 AND OUT3)
        KOUT3         4  (GATHER P-T DATA AT 4 DISTANCES, S(1-4))
        KOUT3A       10  (GATHER P-T DATA EVERY 10 CYCLES)
        KOUT4      4000  (PUNCH CONTINUATION CARDS AT CYCLE 4000)
        KREZ1         0  (REZ1 NOT USED)
        KREZ2         0  (REZ2 NOT USED)
        KREZ3        10  (USE 10 FINE ZONES PER COARSE ZONE)
        KREZ3A       30  (USE A TOTAL OF 30 FINE ZONES)
        KREZ3B       41  (OUTER INTERFACE OF LAST FINE ZONE)
        KREZ4         2  (REDUCE REGION 1 TO 2 ZONES AT TIME TREZO)

CARD 5  NC(12)   ALL ZERO EXCEPT-
        NC(11)        1  (PRINT OVERPRESSURE RATHER THAN ABSOLUTE PR)

CARD 6  NC(13-24) ALL ZERO EXCEPT-
        NC(17)        1  (PRINT INTEGRAL OF D(DX) RATHER THAN ZONE MASS)

CARD 7A II            1  (DATA FOR REGION 1 FOLLOWS)
        NEQST(1)      2  (USE REAL AIR EQUATION OF STATE, EQS2)
        GAMMAI(1) 1.4    (NOT USED. EQS2 COMPUTES GAMMA FOR EACH ZONE)
        NZONES(1)  10    (USE 10 ZONES IN THE FIREBALL)
        OUTBDY(1) 425.1       (OUTER RADIUS OF REGION 1)
        EINIT(1) 1.062E+14    (INTERNAL ENERGY OF MATERIAL IN REGION 1)
        UINIT(1)   0.         (START WITH ALL FIREBALL ZONES AT REST)
        DINIT(1) .001225      (USE AMBIENT SEA LEVEL DENSITY)
        CINQ(1)    2.0   (THE CHOICE OF CINQ AND AINQ IS AN ART. THESE
        AINQ(1)    0.2    VALUES HAVE GIVEN SATISFACTORY RESULTS)
        ZONING(1)  0.    (USE ZONES OF UNIFORM WIDTH)
```

```
CARD 7B II           2  (DATA FOR REGION 2 FOLLOWS)
        NEQST(2)     2  (USE REAL AIR EQUATION OF STATE, EQS2)
        GAMMAI(2) 1.4   (NOT USED)
        NZONES(2)   30  (USE 30 ZONES IN THE FIRST AIR REGION)
        OUTBDY(2) 3500.      (OUTER RADIUS OF REGION 2 IS 35 METERS)
        EINIT(2) 2.068E+9    (ENERGY OF AMBIENT SEA LEVEL AIR)
        UINIT(2)    0.       (START AT REST)
        DINIT(2) .001225     (SEA LEVEL AMBIENT DENSITY)
        CINQ(2)    2.0
        AINQ(2)    0.2
        ZONING(2)   0.

CARD 7C SAME AS CARD 7B EXCEPT-
        II           3  (DATA FOR REGION 3 FOLLOWS)
        NZONES(3) 200   (USE 200 ZONES IN REGION 3)
        OUTBDY(3) 2.E+5  (OUTER BOUNDARY OF PROBLEM IS AT 2000 METERS)

CARD 8  COSPHI      0.  (NOT USED SINCE IARDC=0)
        HOBKM       0.  (CENTER OF EXPLOSION IS AT SEA LEVEL)
        WKT         0.  (NOT USED)
        BLANK1      0.  (NOT USED)
        BLANK2      0.  (NOT USED)
        ZONSIZ      0.  (NOT USED)

CARD 9  T           0.  (START WITH TIME=0.)
        TREZO      .01  (REDUCE REGION 1 TO 2 ZONES AT .01 SECOND)
        DTMIN(2)    0.  (USE BUILT-IN VALUE OF 1.E-10 SECONDS)
        DTRATE      0.  (USE BUILT-IN VALUE OF 1.4)
        STABIL      0.  (USE BUILT-IN VALUE OF 0.801)
        UCUT        0.  (USE BUILT-IN VALUE OF 1.E-8 CM/SEC)

CARD 10 TLIST(1-6)  ALL ZERO (PRINTING IS CONTROLLED BY NPR)

CARD 11 S(1)     1800.  (COLLECT P-T DATA AT 18, 38, 90, AND 250 METERS)
        S(2)     3800.
        S(3)     9000.
        S(4)    25000.
        S(5)        0.  (NOT USED)
        S(6)        0.  (NOT USED)

CARD 12 S(7-12) ALL ZERO (NOT USED)

CARD 13  BLANK EXCEPT FOR 0 IN COLUMN 5 (NO MORE PROBLEMS FOLLOW)
```

THE DATA CARDS FOR THIS PROBLEM MIGHT APPEAR AS FOLLOWS-

```
$DATA
WUNDY CODE CALCULATION FOR 1 KT AT SEA LEVEL.                                    1
 U.S.N.O.L.    3-1-65     1 KT AT SEA LEVEL.                                     2
 1160    3    2    2    3    0    0    0 4000   50    0   20                     3
    4   10    0    4   10 4000    0    0   10   30   41    2                     4
    0    0    0    0    0    0    0    0    0    0    1    0                     5
    0    0    0    0    1    0    0    0    0    0    0    0                     6
 1  2    1.4  10 4.25100+2 1.0620+14       0. 1.22500-3   2.    .2    0.        7A
 2  2    1.4  30      3500. 2.06800+9      0. 1.22500-3   2.    .2    0.        7B
 3  2    1.4 200 2.00000+5 2.06786+9       0. 1.22500-3   2.    .2    0.        7C
         0.         0.         0.         0.         0.         0.               8
         0.        .01         0.         0.         0.         0.               9
         0.         0.         0.         0.         0.         0.              10
      1800.      3800.      9000.     25000.         0.         0.              11
         0.         0.         0.         0.         0.         0.              12
    0
```

THE RUNNING TIME FOR THIS PROBLEM IS 20 MINUTES ON AN IBM 7090.

Some of the possible modifications of this problem are as follows:

1) The program will calculate its own fireball internal energy if EINIT(1)=0. and WKT is positive (here, WKT=1.).

2) To calculate upward from the explosion rather than horizontally, set IARDC=2 to order the ARDC subroutine to provide the ambient conditions in regions 2 and 3 and set COSPHI=1. to indicate an upward ray. To calculate upward at a 45 degree angle, set COSPHI=.707, etc. The ARDC subroutine can of course be used for homogeneous runs, if desired. If IARDC were 2 in section 1) above, EINIT(2), EINIT(3), DINIT(2), and DINIT(3) would be supplied by ARDC and could be zero in the input cards.

3) To calculate downward from 1 KT at 20 km altitude of burst, set WKT=1., IARDC=2, COSPHI=-1., and HOBKM=20.

4) If tape 18 was written during a problem (e.g., by setting NTAPE=500 to cause a record to be written every 500 cycles) the problem can be continued later from any record of this tape. Use the same data cards as originally used to run the problem except for setting NQUIT to some new cycle number to stop on and INTAPE=3 (or whatever record number is desired).

5) To calculate one megaton rather than one kiloton, change the OUTBDY values and WKT. To use the scaling criteria given here (Sachs scaling) simply multiply the OUTBDY values for one kiloton by ten. Actually, for a homogeneous calculation (COSPHI=0.), the 1 KT results could be used directly for any yield by applying the proper scaling factors to all quantities in the output. It is generally more convenient, however, to re-run the problem for each different yield. The results for nonhomogeneity runs (COSPHI non-zero) <u>do not</u> Sachs scale and each yield and altitude of burst must be calculated separately.

6) To calculate the pressure incident on the ground from an air burst, proceed as in 3) above but adjust the zoning so OUTBDY(3) equals the distance to the ground. The shockwave will reflect from this rigid boundary as from the ground, but the problem will become meaningless soon after reflection because the spherical symmetry will force the shock to converge back on the origin rather than behave as a true reflected shock apparently originating from a focus beneath the ground. Calculation of reflected shocks and Mach stems would require a two-dimensional code.

APPENDIX B

LISTINGS FOR WUNDY HYDROCODE

GLOSSARY OF DIMENSIONED QUANTITIES

I = REGION INDEX, RUNS FROM 1 TO 30 OR 31
J = ZONE INDEX, RUNS FROM 1 TO 300 OR 301
* = NORMALLY READ IN AT START OF PROBLEM
** = NOT NORMALLY READ IN BUT CAN BE READ FROM CARDS BY GENSUB

* AINQ(I)	COEFFICIENT IN LINEAR ARTIFICIAL VISCOSITY TERM.
* ALIST(1-12)	ARBITRARY LETTERING TO BE PRINTED AT START.
* ALIST(12-24)	ARBITRARY LETTERING TO BE PRINTED ON EVERY OUTPUT CYCLE.
B(J)	NOT USED.
C(J)	NOT USED.
* CINQ(I)	COEFFICIENT IN QUADRATIC ARTIFICIAL VISCOSITY TERM.
** D(J)	DENSITY (G/CC).
* DINIT(I)	INITIAL VALUE OF D FOR ALL ZONES IN REGION I (G/CC).
DTMIN(1)	TIMESTEP CENTERED AT TIME N-1/2 .
(2)	TIMESTEP CENTERED AT TIME N+1/2 .
(3)	TIMESTEP CENTERED AT TIME N.
DTZJ(J)	TIME STEP ASSOCIATED WITH ZONE J.
DUDT(J)	ACCELERATION OF INTERFACE J.
DUMMY(J)	AUXILIARY VARIABLE, USED BY GENR WHEN READING CARDS.
DWAS(J)	VALUE OF D(J) AT PREVIOUS TIME STEP.
** E(J)	INTERNAL ENERGY OF ZONE J (ERGS/GRAM).
* EINIT(I)	INITIAL TOTAL INTERNAL ENERGY OF REGION I
EINT(I)	TOTAL INTERNAL ENERGY OF REGION I
EKIN(I)	TOTAL KINETIC ENERGY OF REGION I
ENTOT(I)	TOTAL ENERGY (INTERNAL + KINETIC) OF REGION I
EWAS(J)	INTERNAL ENERGY E(J) AT PREVIOUS TIME STEP.
F(J)	NOT USED.
* GAMMAI(I)	GAMMA FOR REGION I. IF GAMMAJ IS CALCULATED FOR EACH ZONE BY EQST, IT SUPERCEDES GAMMAI.
GAMMAJ(J)	GAMMA FOR ZONE J. (EQUATION OF STATE CONSTANT).
GRAMS(J)	MASS OF ZONE J (GRAMS).
INTERJ(I)	NUMBER OF THE INTERFACE FORMING INNER BOUNDARY OF REG I.
JIN(J)	AUXILIARY VARIABLE. USED BY GENR WHEN CARDS ARE READ.
* NC(1-24)	LOGICAL FLOW CONTROL VARIABLES, NORMALLY ZERO.
* NC(1)	=1 LIMIT THE RATE OF Q GROWTH (VERY RARELY USED).
* NC(2)	NOT USED.
* NC(3)	NOT USED.
* NC(4)	=1 SUPPRESS ENERGY CHECK. FOR USE IN PROBLEMS WHERE ENERGY IS NOT CONSERVED (E.G., MOVING PISTONS).
* NC(5)	=1 LEAVE OUT GRAVITY TERM (FOR TESTING SIZE OF GRAVITY EFFECT ON NONHOMOGENEITY PROBLEMS).
* NC(6)	=1 PRINT NC(6) TIMES PER DECADE IN TIME.
* NC(7)	=1 USE SPECIAL Q (FOR PROBLEMS WITH SEVERE MOTIONS NEAR THE ORIGIN).

*	NC(8)	=1 SUPPRESS TRIGGERING OF REZ3 ON P CHANGE.
*	NC(9)	=0 PUT OUT2 OUTPUT ON TAPE 6. =2, TAPE 26.
*	NC(10)	=0 PUT OUT3 OUTPUT ON TAPE 6. =2, TAPE 26.
*	NC(11)	=1 PRINT OVERPRESSURE RATHER THAN ABSOLUTE PSI.
*	NC(12)	=1 DUMP THE MEMORY CORES AFTER PROBLEM IS DONE.
*	NC(13)	=1 CALL SUBROUTINE FIXX EVERY CYCLE.
*	NC(14)	=0 PUT REZONING COMMENTS ON TAPE 6. =2, ON TAPE 26.
*	NC(15)	= USE NC(15) STEPS OF ONE CYCLE IN OUT3.
*	NC(16)	=1 PRINT BEFORE AND AFTER EACH TIME REZ3 IS USED.
*	NC(17)	=1 PRINT CUMULATIVE GRAMS/SQ CM INSTEAD OF PDYN.
*	NC(18)	= DO NOT CARRY MAX Q SEARCH IN OUT2 INTO REGION NC(18).
*	NC(19)	=1 PRINT LIST OF ALL TAPE 18 OUTPUT.
*	NC(20-24)	NOT USED.
	NC(25-40)	INTERNAL FLAGS FOR LOGICAL FLOW CONTROL.
	NC(25)	=1 CALCULATE ONLY THE TIMESTEP IN HYDR (RARELY USED).
	NC(26)	NUMBER OF ACTIVE LOCATIONS IN OUT3.
	NC(27)	NOT USED.
	NC(28-40)	NOT USED.
*	NEQST(I)	NUMBER OF EQUATION OF STATE TO BE USED IN REGION I.
*	NZONES(I)	NUMBER OF ZONES IN REGION I.
*	OUTBDY(I)	INITIAL X-COORDINATE OF OUTER INTERFACE OF REGION I.
	P(J)	PRESSURE BEFORE Q HAS BEEN ADDED (DYNE/SQ CM).
	PDYN(J)	DYNAMIC PRESSURE (PSI).
	PQ(J)	PRESSURE (MEGABARS) =P(J)+Q(J)
	PSI(J)	PQ(J) IN PSI UNITS.
	PWAS(J)	VALUE OF P(J) AT PREVIOUS TIME STEP.
**	Q(J)	ARTIFICIAL VISCOSITY CONTRIBUTION TO PRESSURE.
*	S(L)	SPARE INPUT QUANTITIES. OUT3 USES SOME AS DISTANCES.
	SOUND(J)	SOUND SPEED IN ZONE J.
	TEMP(J)	TEMPERATURE (KELVIN).
*	TLIST(1-8)	PRINTING TIMES, RARELY USED. USUALLY=0.
		PRINT EVERY TLIST(1) SECONDS UNTIL TLIST(2) IS REACHED,
		THEN PRINT EVERY TLIST(3) UNTIL TLIST(4) IS REACHED, AND
		THEN STOP THE CALCULATION. WRITE TV TAPE EVERY TLIST(5)
		SEC UNTIL TLIST(2) AND EVERY TLIST(6) THEREAFTER.
**	U(J)	INTERFACE VELOCITY (CM/SECOND).
*	UINIT(I)	INITIAL VELOCITY OF ALL INTERFACES OF REGION I.
**	X(J)	COORDINATE OF INTERFACE (CM).
	XAV(J)	COORDINATE OF CENTER OF ZONE (CM).
	Y(J)	NOT USED.
*	Z(J)	Z(1-30) ARE READ IN BUT NOT PRESENTLY USED.
*	ZONING(I)	TYPE OF ZONING IN REGION I.
		=0. USE ZONES OF EQUAL WIDTH THROUGHOUT I.
		=POSITIVE. LET THE RATIO OF SIZE OF OUTERMOST ZONE TO INNERMOST ZONE IN I BE ZONING(I) AND FILL I WITH ZONES OF GEOMETRICAL PROGRESSION IN WIDTH.

GLOSSARY OF NON-DIMENSIONED QUANTITIES

* = NORMALLY READ IN AT START OF PROBLEM

	AREA	AREA OF INTERFACE J (SQUARE CENTIMETERS).
*	BLANK1	NOT USED.
*	BLANK2	NOT USED.
	BLANK3	NOT USED.
*	COSPHI	COSINE OF ANGLE GIVING DIRECTION OF RAY BEING CALCULATED, MEASURED FROM ZENITH. +1. FOR UP, 0. FOR HORIZONTAL, -1. FOR DOWN, ETC. USED FOR MULTIPLYING GRAVITY TERM IN NONHOMOGENEOUS ATMOSPHERE CALCS FOR GETTING ZONE ALTITUDE.
*	DTRATE	RATE AT WHICH TIMESTEP IS ALLOWED TO BUILD UP (USU. 1.4).
	EINSUM	TOTAL INTERNAL ENERGY IN ENTIRE PROBLEM.
	EKSUM	TOTAL KINETIC ENERGY IN ENTIRE PROBLEM.
	ESTART	TOTAL ENERGY IN ENTIRE PROBLEM AT BEGINNING (ERGS).
	ETOTAL	TOTAL ENERGY IN ENTIRE PROBLEM AT CURRENT TIME (ERGS).
	GACCEL	COMPONENT OF GRAVITY ALONG RAY BEING CALCULATED(CM/SECSQ).
*	HOBKM	ALTITUDE OF CENTER OF PROBLEM (KILOMETERS).
	I	REGION INDEX, RANGES FROM 1 TO IMAX.
*	IARDC	SET UP WITH ARDC ATMOSPHERE IN REGIONS IARDC TO IMAX. =0 DO NOT USE ARDC AT ALL.
	IDENT	USED FOR IDENTIFYING SUBROUTINES WHEN BEGINNING PROBLEM.
*	IMAX	NUMBER OF REGIONS IN PROBLEM (UP TO 30).
*	INCODS	=0 NO EXTRA DATA CARDS TO BE READ IN GENR. =1 READ TYPE I CARDS. =2 READ TYPE II CARDS.
*	INTAPE	READ INITIAL CONDITIONS FROM RECORD INTAPE OF TAPE 18.
	J	ZONE INDEX. USED IN SEVERAL PLACES.
*	JCALC	NUMBER OF LAST INTERFACE CURRENTLY BEING CALCULATED IN EACH SWEEP. EVERYTHING BEYOND JCALC IS ASSUMED TO BE INACTIVE.
	JCALC1	=JCALC-1
	JCALC2	=JCALC-2
	JDT	INDEX OF ZONE CONTAINING SMALLEST TIME STEP.
	JLAST	NUMBER OF THE LAST INTERFACE IN THE PROBLEM. (UP TO 301).
	JLAST1	=JLAST-1
	JMAX	OUTER INTERFACE OF LAST ZONE OF REGION BEING CALCULATED.
	JMIN	OUTER INTERFACE OF FIRST ZONE OF REGION BEING CALCULATED.
	JMH	INDEX OF CENTER OF ZONE WHOSE OUTER INTERFACE IS J.
	JPH	INDEX OF CENTER OF ZONE WHOSE INNER INTERFACE IS J.
	JUTEST	SOME INTERFACE OUTSIDE THE INITIALLY ACTIVE REGION.
*	K	PROBLEM SYMMETRY. =1 PLANE =2 CYLINDRICAL =3 SPHERICAL
*	KOUT2	=0 DO NOT CALL OUT2. (USUALLY 4 WHEN OUT2 IS USED.)
*	KOUT2A	=L PREPARE A LINE OF SUMMARY EVERY L-TH CYCLE.
*	KOUT2B	=-1 READ IN TRAJECTORY DATA RX,RY,SPEED IN OUT3. =0 USE S(L) LIST AS R(L) DISTANCES IN OUT3. =L (1 TO 8) LET OUT2 LOCATE FOR OUT3 THE R(L) WHERE DESIRED OVERPRESSURE LEVELS S(L) OCCUR AND LET OUT3 ACCUMULATE P-T DATA AT THESE DISTANCES.

* KOUT3	NUMBER OF DISTANCES (OR OVERPRESSURES, IF NC(8) POSITIVE) IN OUT3. OUT3 IS NOT USED IF KOUT3=0.
* KOUT3A	STORE A LINE OF DATA FOR EACH DISTANCE EVERY KOUT3A CYCLES.
* KOUT4	=1 USE OUT4.
* KREZ1	=1 USE OUT1.
* KREZ2	=L USE REZ2 BEGINNING WITH REGION L.
* KREZ3	=0 OMIT REZ3. =L USE L FINE ZONES PER LARGE ZONE.
* KREZ3A	= TOTAL NUMBER OF FINE ZONES.
* KREZ3B	J INDEX OF INTERFACE BOUNDING OUTERMOST FINE ZONE.
* KREZ4	=0 DO NOT USE REZ4. IF NON-ZERO, REDUCE REGION 1 TO KREZ4 ZONES WHEN TIME TREZO IS REACHED.
KUTOFF	=1 TROUBLE. PRINT AND THEN STOP THE PROBLEM.
L	GENERAL SUBSCRIPT, USED IN MANY PLACES.
MAS	=0 NORMAL PRINTOUT.
	=1 PRINT ZONE MASSES INSTEAD OF DYNPSI (USED INITIALLY AND AFTER REZONING).
* MUREX	TYPE OF OUTER BOUNDARY (INTERFACE JLAST) ON PROBLEM.
	=1 FREE SURFACE.
	=2 RIGID WALL.
	=3 BOUNDARY MOTION TO BE SPECIFIED BY SUBROUTINE BDY2.
* MURIN	TYPE OF INNER BOUNDARY (INTERFACE 1) ON PROBLEM.
	=1 FREE SURFACE.
	=2 RIGID WALL.
	=3 BOUNDARY MOTION TO BE SPECIFIED BY SUBROUTINE BDY1.
N	NOT USED.
* NCYCLE	CYCLE NUMBER. EACH CYCLE IS ONE COMPLETE SWEEP THRU MESH.
NDOALL	=1 CALCULATE ALL ZONES WHETHER ACTIVE OR NOT.
NGEOM	SPECIFIES WHAT IS WANTED FROM SUBROUTINE GEOM (Q.V.). =1 CALCULATE AREA, =2 CALCULATE VOLUME.
* NPR	PRINT OUTPUT EVERY NPR CYCLES (USUALLY ABOUT 25).
NPRINT	=1 FORCES PRINTING AFTER CURRENT CYCLE.
* NPROB	ARBITRARY PROBLEM NUMBER.
NQST	=NEQST(I).
* NQUIT	TERMINATE THE PROBLEM WHEN NCYCLE REACHES NQUIT. (WUNDY WILL DO ABOUT 2000 CYCLES FOR 40 ACTIVE ZONES IN ABOUT 10 MINUTES ON THE 7090)
NRECRD	NUMBER OF RECORD ON TAPE 18. (1=FIRST RECORD, ETC.).
* NTAPE	WRITE TAPE 18 EVERY NTAPE CYCLES (USUALLY 500). =0 DO NOT USE TAPE 18.
NWRITE	FORCES WRITING OF TAPE 18 AFTER CURRENT CYCLE.
* STABIL	STABILITY CONSTANT (USUALLY ABOUT 0.8).
* T	CURRENT TIME (SECONDS).
* TREZO	CALL REZ4 WHEN TIME TREZO IS REACHED.
* UCUT	SET ALL INTERFACE VELOCITIES LESS THAN UCUT EQUAL TO 0. USE UCUT=1.E-2 IF UCUT=0. IN THE INPUT.
USQ	SQUARE OF INTERFACE VELOCITY.
VOL	VOLUME OF ZONE BOUNDED BY INTERFACES J AND J-1 .

```
* WKT          FIREBALL ENERGY, KILOTONS.  IF THIS IS NON-ZERO, PROGRAM
               SUPPLIES VALUES FOR ANY OF THE FOLLOWING IF THEY ARE
               ZERO IN THE INPUT--
               NZONES(1-3),OUTBDY(1-3),TREZO,DINIT(1),EINIT(1)
* ZONSIZ       AMBIENT ZONE SIZE (METERS) FOR 1 KT AT SEA LEVEL.  USED
               BY PROGRAM WHEN IT SETS UP ITS OWN ZONING BY SACHS
               SCALING.  (PROGRAM USES .5 IF ZONSIZ=0.).
```

LIST OF SUBROUTINES

```
   MAIN        HANDLES MAIN LOGICAL FLOW.
   GENR        READS INPUT AND INITIALIZES THE PROBLEM.
   HYDR        COMPUTES HYDRODYNAMIC MOTIONS AND NEXT TIMESTEP.
   OUT1        PRINTS NORMAL OUTPUT ON TAPE 6 AND WRITES TAPE 18.
   OUT2        ACCUMULATES DATA ON MAIN SHOCK FRONT POSITION, ETC.
   OUT3        ACCUMULATES PRESSURE, ETC VS TIME DATA AT FIXED POSITIONS.
   OUT4        PUNCHES CARDS FROM WHICH THE PROBLEM MAY BE CONTINUED.
   REZ1        REMOVES EXCESSIVELY COMPRESSED ZONES.
   REZ2        EXPANDS RANGE OF PROBLEM BY FACTOR OF TWO.
   REZ3        KEEPS FINE ZONES IN SHOCK.
   REZ4        REZONES FIRST REGION AT TIME TREZO.
   ARDC        CONTAINS ARDC MODEL ATMOSPHERE.
   GEOM        DESCRIBES PROBLEM GEOMETRY.
   EQST        CONTROLS ACCESS TO EQUATIONS OF STATE.
   EQS1        EQUATION OF STATE FOR GAMMA LAW GAS, E=PV/(G-1).
   EQS2        APPROXIMATE EQUATION OF STATE FOR REAL AIR.
** EQS3-12     RESERVED FOR OTHER EQUATIONS OF STATE.
** BDY1        SPECIFIES MOTION OF INNER INTERFACE.
** BDY2        SPECIFIES MOTION OF OUTER INTERFACE.
** FIXX        USED FOR MAKING SPECIAL CHANGES DURING A RUN.

** NOT INCLUDED IN THESE LISTINGS.
```

UNUSED SUBROUTINES MAY BE REPLACED BY DUMMY DECKS, E.G.,

```
$IBFTC BDY2    LIST,REF
      SUBROUTINE BDY2
      NULL=0
      RETURN
      END
```

STANDARD COMMON STATEMENT

```
 COMMON       AINQ(30)    ,ALIST(24)   ,B(300)      ,C(300)
1,CINQ(30)   ,D(300)      ,DINIT(30)   ,DTMIN(3)    ,DTZJ(300)
2,DUDT(301)  ,DUMMY(301)  ,DWAS(300)   ,E(300)      ,EINIT(30)
3,EINT(30)   ,EKIN(30)    ,ENTOT(30)   ,EWAS(300)   ,F(300)
4,GAMMAI(30) ,GAMMAJ(300),GRAMS(300)  ,INTERJ(31)  ,JIN(301)
5,NC(40)     ,NEQST(30)   ,NZONES(30)  ,OUTBDY(30)  ,P(300)
6,PDYN(300)  ,PQ(300)     ,PSI(300)    ,PWAS(300)   ,Q(300)
7,S(24)      ,SOUND(300)  ,TEMP(300)   ,TLIST(8)    ,U(301)
8,UINIT(30)  ,X(301)      ,XAV(300)    ,Y(301)      ,Z(301),ZONING(30)
 COMMON  AREA  ,BLANK1,BLANK2,BLANK3,COSPHI,DTRATE,EINSUM,EKSUM
1,ESTART,ETOTAL,GACCEL,HOBKM ,I     ,IARDC ,IDENT ,IMAX  ,INCODS
2,INTAPE,J     ,JCALC ,JCALC1,JCALC2,JDT   ,JLAST ,JLAST1,JMAX
3,JMIN  ,JMH   ,JPH   ,JUTEST,K     ,KOUT2 ,KOUT2A,KOUT2B,KOUT3
4,KOUT3A,KOUT4 ,KREZ1 ,KREZ2 ,KREZ3 ,KREZ3A,KREZ3B,KREZ3C,KREZ4
5,KUTOFF,L     ,MAS   ,MUREX ,MURIN ,N     ,NCYCLE,NDOALL,NGEOM
6,NPR   ,NPRINT,NPROB ,NQST  ,NQUIT ,NRECRD,NTAPE ,NTVREC,NWRITE
7,STABIL,T     ,TREZO ,UCUT  ,USQ   ,VOL   ,WKT   ,ZONSIZ
```

```
$IBFTC MAIN    LIST,REF,DD
C     MAIN PROGRAM                                  1-D HYDROCODE  MAIN
C     NOL ONE-DIMENSIONAL HYDRODYNAMIC CODE.
C     THIS PROGRAM IS SET UP FOR CM-GRAM-SECOND UNITS.
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
    1 IDENT=0
          CALL GENR
C
C       BEGIN IDENTIFICATION SECTION.
      WRITE(6,5)                                                       *
    5 FORMAT(40H1MAIN PROGRAM    VERSION  3,  3- 1-65    )
      IDENT=1
          CALL GENR
          CALL HYDR
          CALL OUT1
      IF(KOUT2.GT.0) CALL OUT2
      IF(KOUT3.GT.0) CALL OUT3
      IF(KOUT4.GT.0) CALL OUT4
      IF(KREZ1.GT.0) CALL REZ1
      IF(KREZ2.GT.0) CALL REZ2
      IF(KREZ3.GT.0) CALL REZ3
      IF(KREZ4.GT.0) CALL REZ4
      IF(NC(13).GT.0) CALL FIXX
          CALL GEOM
   20 DO 22 I=1,IMAX
      NQST=NEQST(I)
          CALL EQST
   22 CONTINUE
      IDENT=0
C       END IDENTIFICATION SECTION.
C
C       PRINT INITIAL CONDITIONS.
      NPRINT=1
      MAS=1
          CALL OUT1
C
C       BEGIN MAIN CALCULATION LOOP.
C       CHECK IF PROBLEM IS COMPLETED YET.
   25 IF(T.GE.TLIST(4)) GO TO 50
   26 IF(NQUIT) 50,28,27
   27 IF(NCYCLE.GE.NQUIT) GO TO 50
C       STOP PROBLEM IF ZERO TIMESTEP OCCURS.
   28 IF(DTMIN(2).GT.0.) GO TO 31
   29 WRITE(6,30)
   30 FORMAT(18H0TIMESTEP IS BAD. )
      GO TO 50
   31 T=T+DTMIN(2)
      NCYCLE=NCYCLE+1
          CALL HYDR
```

```
         CALL OUT1                                                    MAIN
C       RESET CALCULATE-EVERY-ZONE-REGARDLESS INDICATOR.
C     REMOVE THE FOLLOWING STATEMENT IF ALL ZONES ARE TO BE CALCULATED
C     THROUGHOUT THE PROBLEM.
      NDOALL=0
      IF(KOUT2.GT.0) CALL OUT2
      IF(KOUT3.GT.0) CALL OUT3
      IF(KOUT4.GT.0) CALL OUT4
      IF(KREZ1.GT.0) CALL REZ1
      IF(KREZ2.GT.0) CALL REZ2
      IF(KREZ3.GT.0) CALL REZ3
      IF(KREZ4.GT.0) CALL REZ4
      IF(NC(13).GT.0) CALL FIXX
      GO TO 25
C       END MAIN CALCULATION LOOP.
C
   50 WRITE(6,51)                                                     *
   51 FORMAT(40H0   NORMAL END REACHED IN MAIN ROUTINE  )
      NPRINT=1
      NWRITE=1
         CALL OUT1
C
C       SETTING NQUIT NEGATIVE CAUSES FINAL PRINTOUTS.
      NQUIT=-1
      IF(KOUT2.GT.0) CALL OUT2
      IF(KOUT3.GT.0) CALL OUT3
      IF(KOUT4.GT.0) CALL OUT4
      IF(NC(13).GT.0) CALL FIXX
C       WRITE END-OF-FILE ON TAPE 26 (IF USED).
      IF(NC(9).EQ.2.OR.NC(10).EQ.2) END FILE 26
C       PRINT TERMINATION CONDITIONS.
   57 WRITE(6,58)                                                     *
   58 FORMAT(33H1PROBLEM CONDITIONS AT END OF RUN)
      WRITE(6,59)    IMAX,KREZ3B,JCALC,NRECRD                         *
   59 FORMAT(6H0IMAX=I2,10H   KREZ3B=I3,10H    JCALC=I3,10H   NRECRD=I3)
      WRITE(6,60)    (NZONES(L),L=1,IMAX)                             *
      WRITE(6,63)  NCYCLE,T                                           *
   63 FORMAT(8H0NCYCLE=I5,8H   TIME=1PE11.3)
   60 FORMAT(8H0NZONES=15I4)
C       DO ANOTHER PROBLEM IF NEXT=1
   61 READ(5,62) NEXT                                                 *
   62 FORMAT(I5)
   99 IF(NC(12).GT.0) CALL PDUMP(AINQ(1),ZONSIZ,1,AINQ(1),ZONSIZ,2)
      IF(NEXT.EQ.1) GO TO 1
      STOP
       END
```

```
$IBFTC GENR    LIST,REF
      SUBROUTINE GENR
C     INPUT GENERATOR                                                   GENR
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
    1 FORMAT(12A6)
    2 FORMAT(12I5)
    3 FORMAT(I2,I3,F6.3,I4,4E10.5,3F5.2)
    4 FORMAT(6E10.4)
    5 FORMAT(53H1  PRINTOUT OF INITIAL CONSTANTS                    )
    6 FORMAT(1H0,12A6)
    8 FORMAT(96H0  NPROB       K    MURIN   MUREX    IMAX   IARDC  INTAPE
     1  INCODS   NQUIT     NPR   NTAPE   JCALC   )
    9 FORMAT(12I8)
   10 FORMAT(96H0  KOUT2  KOUT2A  KOUT2B   KOUT3  KOUT3A   KOUT4   KREZ1
     1   KREZ2   KREZ3  KREZ3A  KREZ3B   KREZ4      )
   11 FORMAT(96H0  NC(1)   NC(2)   NC(3)   NC(4)   NC(5)   NC(6)   NC(7)
     1   NC(8)   NC(9)  NC(10)  NC(11)  NC(12))
   12 FORMAT(96H0 NC(13)  NC(14)  NC(15)  NC(16)  NC(17)  NC(18)  NC(19)
     1  NC(20)  NC(21)  NC(22)  NC(23)  NC(24))
   13 FORMAT(105H0I  NEQST  GAMMA  NZONES   OUTBDY        EINIT
     1UINIT          DINIT         CINQ  AINQ ZONING       )
   14 FORMAT(I2,I7,F7.3,I8,1P4E14.5,0P3F6.2)
   15 FORMAT(96H0  COSPHI                HOBKM              WKT          BLA
     1NK1           BLANK2             ZONSIZ             )
   16 FORMAT(96H0     T                  TREZO            DTMIN(2)        DTR
     1ATE           STABIL             UCUT               )
   17 FORMAT(1P6E16.5)
   18 FORMAT(10H0S(L) LIST)
   19 FORMAT(15H0PRINTING TIMES)
   20 FORMAT(5(I4,E10.3))
C
      IF(IDENT.NE.1) GO TO 30
   28 WRITE(6,29)                                                       *
   29 FORMAT(40H0SUBROUTINE GENR VERSION  3,  3- 1-65    )
      RETURN
C
   30 READ(5,1)     (ALIST(L),L=1,12)                                   *
      READ(5,1)     (ALIST(L),L=13,24)                                  *
      READ(5,2)     NPROB ,K     ,MURIN ,MUREX ,IMAX  ,IARDC ,          *
     1              INTAPE,INCODS,NQUIT ,NPR   ,NTAPE ,JCALC            *
      READ(5,2)     KOUT2 ,KOUT2A,KOUT2B,KOUT3 ,KOUT3A,KOUT4 ,          *
     1              KREZ1 ,KREZ2 ,KREZ3 ,KREZ3A,KREZ3B,KREZ4            *
      READ(5,2)     (NC(L),L=1,12)                                      *
      READ(5,2)     (NC(L),L=13,24)                                     *
      READ(5,3)     (II,NEQST(I),GAMMAI(I),NZONES(I),OUTBDY(I),EINIT(I), *
     1              UINIT(I),DINIT(I),CINQ(I),AINQ(I),ZONING(I),I=1,IMAX)*
      READ(5,4)     COSPHI,HOBKM,WKT    ,BLANK1,BLANK2,ZONSIZ           *
      READ(5,4)     T,TREZO,DTMIN(2),DTRATE,STABIL,UCUT                 *
      READ(5,4)     (TLIST(I),I=1,6)                                    *
```

```
      READ(5,4)     (S(L),L=1,12)                                            *
      KUMBAK=1                                                               GENR
C
      WRITE(6,5)                                                             *
   31 WRITE(6,6)    (ALIST(L),L=1,12)                                        *
      WRITE(6,6)    (ALIST(L),L=13,24)                                       *
      WRITE(6,8)                                                             *
      WRITE(6,9)    NPROB ,K      ,MURIN ,MUREX ,IMAX   ,IARDC ,             *
     1              INTAPE,INCODS,NQUIT ,NPR    ,NTAPE ,JCALC                *
      WRITE(6,10)                                                            *
      WRITE(6,9)    KOUT2 ,KOUT2A,KOUT2B,KOUT3 ,KOUT3A,KOUT4 ,               *
     1              KREZ1 ,KREZ2 ,KREZ3 ,KREZ3A,KREZ3B,KREZ4                 *
      WRITE(6,11)                                                            *
      WRITE(6,9)    (NC(L),L=1,12)                                           *
      WRITE(6,12)                                                            *
      WRITE(6,9)    (NC(L),L=13,24)                                          *
      WRITE(6,13)                                                            *
      WRITE(6,14)   (I ,NEQST(I),GAMMAI(I),NZONES(I),OUTBDY(I),EINIT(I),     *
     1              UINIT(I),DINIT(I),CINQ(I),AINQ(I),ZONING(I),I=1,IMAX)    *
      WRITE(6,15)                                                            *
C       PRINT ZERO INSTEAD OF HOBKM AND WKT.
      ZERO=-0.
      WRITE(6,17)    COSPHI,ZERO   ,ZERO ,BLANK1,BLANK2,ZONSIZ               *
      WRITE(6,16)                                                            *
      WRITE(6,17)    T,TREZO,DTMIN(2),DTRATE,STABIL,UCUT                     *
      WRITE(6,19)                                                            *
      WRITE(6,17)    (TLIST(L),L=1,6)                                        *
      WRITE(6,18)                                                            *
      WRITE(6,17)    (S(L),L=1,12)                                           *
      GO TO (32,83),KUMBAK
C
C       MAKE VARIOUS ADJUSTMENTS OF THE INPUT.
C       VELOCITIES LESS THAN UCUT WILL BE SET TO ZERO.
   32 IF(UCUT.LE.0.) UCUT=.01
      IF(DTMIN(2).LE.0.) DTMIN(2)=1.E-10
   36 DTMIN(3)=DTMIN(2)/2.
      IF(DTRATE.LE.0.) DTRATE=1.4
      IF(STABIL.LE.0.) STABIL=.801
      IF(IARDC.LE.0)  IARDC=31
C
C       THE SECTION FROM 475 THRU 501 IS USED ONLY FOR AUTOMATICALLY
C       SETTING UP UNIFORM HOT-AIR-SPHERE EXPLOSION CALCULATIONS.
C       IT IS BYPASSED IF WKT=0.
C       WKT   IS YIELD IN KILOTONS.
C       ZONSIZ IS ZONE SIZE FOR 1KT/SL AND IS ABOUT .5 TO 1.
C       THE FOLLOWING SECTION (475 THRU 501) REPLACES ZEROES IN NZONES,
C       OUTBDY, EINIT, AND DINIT(1) BY SACHS SCALED VALUES.
  475 IF(WKT.LE.0.) GO TO 50
  479 HOBCM=HOBKM*1.E+5
          CALL ARDC(HOBCM,PRATIO,TRATIO,DRATIO)
```

```
      THIRD=1./3.                                                      GENR
      RSCALE=(1./PRATIO)**THIRD
      WSCALE=WKT**THIRD
      EAMB=PRATIO*1.01325E+6/(DRATIO*1.225E-3*.4)
      IF(ZONSIZ.LE.0.) ZONSIZ=.5
  481 ZSIZE=ZONSIZ*100.*RSCALE*WSCALE
      IF(OUTBDY(1).LE.0.) OUTBDY(1)=425.1*RSCALE*WSCALE
      IF(NZONES(1).LE.0) NZONES(1)=10
      IF(NZONES(2).LE.0) NZONES(2)=30
      ZN2=NZONES(2)
      IF(OUTBDY(2).LE.0.) OUTBDY(2)=ZSIZE*ZN2+OUTBDY(1)
  488 NZ3LIM=299-NZONES(1)-NZONES(2)
      IF(NZONES(3).GT.0) GO TO 4890
  489 NZONES(3)=NZ3LIM
      GO TO 490
 4890 IF(OUTBDY(3).GT.0.) GO TO 496
  490 ZN3=NZONES(3)
      ZNELL=KREZ3
      DX3=ZSIZE*ZNELL
      IF(OUTBDY(3).GT.0.) GO TO 492
  491 OUTBDY(3)=DX3*ZN3+OUTBDY(2)
      GO TO 496
  492 SIZE3=OUTBDY(3)-OUTBDY(2)
      NZ3=SIZE3/DX3
      IF(NZ3.LE.NZ3LIM) GO TO 496
  494 NZONES(3)=NZ3
      WRITE(6,495)   NZ3,NZ3LIM                                        *
  495 FORMAT(33H REGION 3 HAD TO BE ADJUSTED FROM,I4,3H TO,6H ZONES)
  496 IF(TREZO.LE.0.) TREZO=.01*RSCALE*WSCALE
      IF(DINIT(1).LE.0.) DINIT(1)=DRATIO*.001225
  501 IF(EINIT(1).GT.0.) GO TO 50
      EINIT(1)=WKT*4.185E19/(4.1887902*OUTBDY(1)**3*DINIT(1)) +EAMB
C       END SPECIAL INPUT CALC SECTION.
C
C       LEAVE OUT GRAVITY IF NC(5) IS POSITIVE.
   50 GACCEL=0.
      IF(NC(5).LE.0) GACCEL=COSPHI*980.616
   53 DO 54 L=25,40
   54 NC(L)=0
      NCYCLE=0
C
   58 NRECRD=0
      JDT=0
      KUTOFF=0
C       FORCE CALC OF ALL ZONES ON FIRST PASS IN HYDR.
      NDOALL=1
      MAS=1
      IF(TLIST(1).EQ.0.) TLIST(1)=1.E20
      IF(TLIST(4).GT.0.) GO TO 61
   59 DO 60 L=2,6
```

```
   60 TLIST(L)=1.E20                                                   GENR
   61 TLIST(7)=T+TLIST(1)
C
C        CALCULATE REGION INTERFACE NUMBERS.
   62 INTERJ(1)=1
      DO 63 I=1,IMAX
      INTERJ(I+1)=INTERJ(I)+NZONES(I)
   63 CONTINUE
C
C        INITIALIZE VARIOUS QUANTITIES.
      JLAST=INTERJ(IMAX+1)
      JLAST1=JLAST-1
      IF(JCALC.EQ.0) JCALC=JLAST
      JCALC1=JCALC-1
      X(1)=0.
      U(1)=0.
      Q(1)=0.
      DTZJ(1)=0.
C
C        CALCULATE INITIAL D AND E FOR EACH ZONE
      DO 76 I=1,IMAX
      JMIN=INTERJ(I)+1
      JMAX=INTERJ(I+1)
   64 IF(ZONING(I)) 67,67,65
   65 XIN=OUTBDY(I-1)
      XOUT=OUTBDY(I)
      LL=NZONES(I)-1
      ZLL=LL
      ZONFAC=ZONING(I)**(1./ZLL)
      SUM=1.
      TERM=1.
      DO 66 L=1,LL
      TERM=TERM*ZONFAC
   66 SUM=SUM+TERM
      ZSIZE=(XOUT-XIN)/SUM
      DELFAC=1.
      GO TO 70
C
   67 DNZONE=NZONES(I)
      IF(I.LE.1) GO TO 69
   68 DELX=(OUTBDY(I)-OUTBDY(I-1))/DNZONE
      GO TO 70
   69 DELX=(OUTBDY(1)-X(1))/DNZONE
C        BEGIN LOOP FOR ZONES WITHIN REGION I.
   70 DO 75 J=JMIN,JMAX
      JMH=J-1
      U(J)=UINIT(I)
      IF(ZONING(I).LE.0.) GO TO 72
   71 X(J)=X(J-1)+DELFAC*ZSIZE
      DELFAC=DELFAC*ZONFAC
```

```
      GO TO 720                                                      GENR
   72 X(J)=X(J-1)+DELX
  720 Q(J)=0.
      P(J)=0.
      DTZJ(J)=0.
      IF(I.LT.IARDC) GO TO 74
C       USE ARDC ATMOSPHERE BEGINNING WITH REGION IARDC
   73 ALTCM=COSPHI*(X(J)+X(J-1))*.5 +HOBKM*1.E+5
          CALL ARDC(ALTCM,PRATIO,TRATIO,DRATIO)
      D(JMH)=DRATIO*.001225
      E(JMH)=PRATIO*1.01325E6/D(JMH)/.4
      GO TO 75
C
   74 D(JMH)=DINIT(I)
      E(JMH)=EINIT(I)
   75 CONTINUE
   76 CONTINUE
C
C       BEGIN AUXILIARY INPUT SECTION
C
   77 IF(INTAPE.LE.0) GO TO 146
C
C       READ BINARY TAPE 18 IF INTAPE IS POSITIVE.
C       START PROBLEM FROM RECORD NUMBER INTAPE OF TAPE 18.
   78 NSKIPA=INTAPE-1
      IF(NSKIPA.LE.0) GO TO 81
   79 REWIND 18
      DO 80 JJ=1,NSKIPA
      READ (18)                                                         *
   80 READ (18)                                                         *
C
   81 READ (18)      T,(DTMIN(L),L=1,3),NCYCLE,NRECRD,JLAST,JCALC,JCALC1,*
     1 KREZ3B,IMAX,NPROB                                                *
      JCALC1=JCALC-1
      JLAST1=JLAST-1
      READ (18)      (NZONES(I),INTERJ(I+1),I=1,IMAX),X(JLAST),U(JLAST), *
     1(X(L),U(L),D(L),E(L),Q(L),PQ(L),TEMP(L),DTZJ(L),L=1,JLAST1)       *
      N18=18
C
  144 WRITE(6,145)    INTAPE,N18                                        *
  145 FORMAT(39H0INITIAL CONDITIONS READ IN FROM RECORD,I3,8H OF TAPE,
     1I3)
C
C       READ TYPE 1 CARDS IF INCODS =1.
C
  146 IF(INCODS.NE.1) GO TO 199
      NAMELIST /LIST1/ B,C,D,DTMIN,DTZJ,DUDT,DUMMY,DWAS,E,EWAS,F,
     1 INTERJ,JIN,NC,NEQST,NZONES,P,PQ,PWAS,Q,TEMP,U,X,Y,Z,
     2 BLANK3,IMAX,INCODS,JCALC,JLAST,JUTEST,KREZ3B,
     3 NCYCLE,NPRINT,NPROB,NRECRD,NWRITE,T
```

```
  147 READ(5,LIST1)
C                                                            *
C       READ TYPE 2 CARDS IF INCODS=2.                       GENR
C
  199 IF(INCODS.NE.2) GO TO 810
  200 READ(5,201) L,X(L),U(L),D(L),E(L),Q(L)
  201 FORMAT(I4,4E14.7,1E12.5)                               *
      IF(L.LT.300) GO TO 200
  202 WRITE(6,203)
  203 FORMAT(38H0INITIAL CONDITIONS READ IN FROM CARDS)      *
C
C       END AUXILIARY INPUT SECTION
C
  810 KUMBAK=2
      WRITE(6,82)
   82 FORMAT(29H1INPUT DATA AFTER ADJUSTMENTS)               *
      GO TO 31
   83 DO 84 L=1,JLAST1
      DWAS(L)=D(L)
      EWAS(L)=E(L)
   84 CONTINUE
C
      DO 87 I=1,IMAX
      I=I
      NQST=NEQST(I)
      JMIN=INTERJ(I)+1
      JMAX=INTERJ(I+1)
      DO 86 J=JMIN,JMAX
      J=J
      JPH=J
      JMH=J-1
   85 NGEOM=2
          CALL GEOM
      GRAMS(JMH)=D(JMH)*VOL
C     EQST PROVIDES P(J),E(J),TEMP(J),GAMMAJ(J), AND SOUND(J).
      NQST=NEQST(I)
          CALL EQST
      PQ(JMH)=P(JMH)+Q(JMH)
   86 CONTINUE
   87 CONTINUE
C
   88 EINSUM=0.
      EKSUM=0.
      ESTART=0.
      DO 90 I=1,IMAX
      EINT(I)=0.
      EKIN(I)=0.
      JMIN=INTERJ(I)
      JMAX=INTERJ(I+1)-1
      DO 89 J=JMIN,JMAX
```

```
      JPH=J
      USQ=(U(J)**2 +U(J+1)**2)*.5
      EINT(I)=EINT(I)+F(JPH)*GRAMS(JPH)
      EKIN(I)=EKIN(I) +.5*USQ*GRAMS(JPH)
   89 CONTINUE
C
      EINSUM=EINSUM+EINT(I)
      EKSUM=EKSUM+EKIN(I)
   90 CONTINUE
      ESTART=EINSUM+EKSUM
      ETOTAL=ESTART
      JCALC1=JCALC-1
      JLAST1=JLAST-1
      JUTEST=JCALC
      RETURN
       END
```

```
$IBFTC HYDR    LIST,REF
      SUBROUTINE HYDR                                                   HYDR
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
      IF(IDENT.NE.1) GO TO 203
  201 WRITE(6,202)                                                      *
  202 FORMAT(40H0SUBROUTINE HYDR VERSION  3,  3- 1-65    )
      KU=0
      RETURN
C
C       CALCULATE ONLY THE TIMESTEP IF NC(25)=1 (NOT NOW USED).
  203 IF(NC(25).GT.0) GO TO 260
C
C       CALCULATE MOTION OF FIRST INTERFACE
  205 J=1
      JPH=1
      MURIN=MURIN
      GO TO (206,207,208),MURIN
  206 NGEOM=1
          CALL GEOM
      DUDT(1)=-PQ(JPH)/GRAMS(JPH)/2. * AREA
      GO TO 210
  207 DUDT(1)=0.
      GO TO 210
  208     CALL BDY1
  210 U(1)=U(1)+DTMIN(3)*DUDT(1)
      X(1)=X(1)+DTMIN(2)*U(1)
C
C       BEGIN MAIN HYDRODYNAMIC LOOP
  211 JDONE=0
      DO 250 I=1,IMAX
      I=I
      JMIN=INTERJ(I)+1
      JMAX=INTERJ(I+1)
      DO 240 J=JMIN,JMAX
      J=J
      JPH=J
      JMH=J-1
C       CALCULATE LAST INTERFACE SEPARATELY
      IF(J.LT.JLAST) GO TO 216
  212 MUREX=MUREX
      GO TO (213,214,215),MUREX
  213 NGEOM=1
          CALL GEOM
      DUDT(J)=PQ(JMH)/GRAMS(JMH)/2. *AREA
      GO TO 220
  214 DUDT(J)=0.
C       GACCEL TERM CAN CAUSE MOTION OF LAST INTERFACE EVEN WHEN DUDT=0.
C       FIX THIS BY TEMPORARILY SETTING GACCEL=0.
```

```
      GSAVE=GACCEL                                                    HYDR
      GACCEL=0.
      GO TO 220
  215     CALL BDY2
      GO TO 220
C
  216 NGEOM=1
          CALL GEOM
      DUDT(J)=(PQ(JMH)-PQ(JPH))/(GRAMS(JMH)+GRAMS(JPH))*2.*AREA-GACCEL
C       ADVANCE U FROM TIME N-1/2 TO TIME N+1/2
  220 U(J)=U(J)+DTMIN(3)*DUDT(J)
C       ADVANCE X FROM TIME N TO TIME N+1
      X(J)=X(J)+DTMIN(2)*U(J)
      NGEOM=2
          CALL GEOM
      PWAS(JMH)=P(JMH)
      EWAS(JMH)=E(JMH)
      DWAS(JMH)=D(JMH)
C       ADVANCE D FROM TIME N TO TIME N+1
      D(JMH)=GRAMS(JMH)/VOL
  223 IF(UCUT.LE.0.) GO TO 226
C       SET VELOCITIES LESS THAN UCUT EQUAL TO ZERO
  224 IF(ABS(U(J)).LT.UCUT) U(J)=0.
C       NDOALL=1 FORCES CALCULATION OF EVERY ZONE IN THE PROBLEM.
  226 IF(NDOALL.EQ.1) GO TO 231
  227 IF(J.LT.JCALC-4) GO TO 231
  228 IF(U(J).NE.0.) KU=-1
  229 KU=KU+1
C
C       CALCULATE ARTIFICIAL VISCOSITY, Q.
C       SAVE PREVIOUS VALUE OF Q
  231 QWAS=Q(JMH)
      IF(ABS((D(JMH)-DWAS(JMH))/DWAS(JMH)).GT.2.E-8) GO TO 2320
      D(JMH)=DWAS(JMH)
      GO TO 238
 2320 IF(D(JMH).LE.DWAS(JMH)) GO TO 238
  232 VOLSUM=1./D(JMH)+1./DWAS(JMH)
      XDIF=X(J)-X(J-1)
 2321 IF(NC(7).GT.0) GO TO 2323
C       USUAL FORM OF VELOCITY GRADIENT.
 2322 UGRAD=U(J-1)-U(J)
      IF(UGRAD.LT.0.) GO TO 238
      GO TO 2324
C       ALTERNATE FORM OF VELOCITY GRADIENT.
 2323 UGRAD=XDIF/DTMIN(3)*(1.-DWAS(JMH)/D(JMH))
 2324 IF(CINQ(I).LE.0.) GO TO 234
  233 Q(JMH)=2.*CINQ(I)**2*UGRAD**2/VOLSUM
      IF(AINQ(I).LE.0.) GO TO 239
      GO TO 235
  234 Q(JMH)=0.
```

```
  235 QLIN=AINQ(I)*UGRAD/VOLSUM*SOUND(JMH)*2.                              HYDR
  237 Q(JMH)=Q(JMH)+QLIN
      GO TO 239
  238 Q(JMH)=0.
C        DO NOT CALCULATE MORE ZONES IF THE LAST 4 ZONES HAD ZERO U.
C        KU COUNTS THE CONSECUTIVE ZONES OF ZERO U.
  239 NQST=NEQST(I)
          CALL EQST
      PQ(JMH)=P(JMH)+Q(JMH)
      IF(JUTEST.GE.J) GO TO 240
      IF(NDOALL.EQ.1) GO TO 240
      IF(KU.LT.4) GO TO 240
      KU=0
      JCALC=J
      JCALC1=J-1
      JMAX=J
      GO TO 251
  240 CONTINUE
  250 CONTINUE
C        END MAIN HYDRODYNAMIC LOOP
C
C        NEXT STATEMENT IS REACHED ONLY WHEN ALL ZONES WERE CALC.
      IF(NDOALL.EQ.1) GO TO 251
      JCALC=JLAST
      JCALC1=JLAST-1
C
C        RESET GACCEL TO ITS PROPER VALUE
  251 GACCEL=GSAVE
C
C        BEGIN TIMESTEP CALCULATION
C
  260 DO 280 I=1,IMAX
      I=I
      JMIN=INTERJ(I)+1
      JMAX=INTERJ(I+1)
      DO 275 J=JMIN,JMAX
      JMH=J-1
      IF(J.GT.JCALC) GO TO 281
  262 UDIF=ABS (U(J)-U(J-1))
      DENOM=SOUND(JMH) +4.*CINQ(I)**2*UDIF +2.*AINQ(I)*SOUND(JMH)
C        STABIL IS A STABILITY CONSTANT (ABOUT .8).
      DTZJ(JMH)=STABIL*(X(J)-X(J-1))/DENOM
  275 CONTINUE
  280 CONTINUE
C
C        FIND ZONE WITH SMALLEST TIMESTEP.
  281 JDT=1
      DO 283 J=2,JCALC1
      IF(DTZJ(J).LT.DTZJ(JDT)) JDT=J
  283 CONTINUE
```

```
      DTNEW=DTZJ(JDT)                                                 HYDR
C         SAVE PREVIOUS VALUE OF TIMESTEP
  284 DTMIN(1)=DTMIN(2)
C         NEW TIMESTEP WILL BE USED TO ADVANCE X.
C         DO NOT LET DTMIN(2) EXCEED DTRATE TIMES PREVIOUS TIMESTEP.
      DTMIN(2)=AMIN1(DTNEW,DTRATE*DTMIN(1))
C         AVERAGE OF OLD AND NEW TIMESTEPS WILL BE USED TO ADVANCE U.
      DTMIN(3)=(DTMIN(2)+DTMIN(1))/2.
C
C         END TIMESTEP CALCULATION
C
  299 RETURN
       END
```

```
$IBFTC OUT1    LIST,REF
      SUBROUTINE OUT1                                                   OUT1
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
      DATA NOUGHT,MCY,NTYMES/0,0,0/
C
      IF(IDENT.NE.1) GO TO 324
  301 WRITE(6,302)                                                     *
  302 FORMAT(40HOSUBROUTINE OUT1 VERSION  3,  3- 1-65    )
      ZNC6=NC(6)
      TFACTR=10.**(1./ZNC6)
      NOUGHT=0
      RETURN
C
  303 FORMAT(1H1,17HPROBLEM NOL WUNDY,I5,2H  ,12A6)
  304 FORMAT(I5,1P6E13.5,2E8.1,E9.2,E12.5)
  305 FORMAT(I6,1P3E14.5,I7)
  306 FORMAT(119HO  J  X(J+1)      U(J+1)     X CENTER  DENSITY   ENERGY
     1   PQ(J)     Q(J)    TOTPSI OR  ZONE MASS    TEMP    DTZJ   GAMMA)
  307 FORMAT(119H         CM         CM/SEC        CM       GM/CC    ERG/GM
     1  DYNE/SQCM DYNE/SQCM OVERPSI     GRAMS     KELVIN     SEC         )
  308 FORMAT(119HO  J  X(J+1)      U(J+1)     X CENTER  DENSITY   ENERGY
     1   PQ(J)     Q(J)    TOTPSI OR  DYNPSI OR    TEMP    DTZJ   GAMMA)
  309 FORMAT(119H         CM         CM/SEC        CM       GM/CC    ERG/GM
     1  DYNE/SQCM DYNE/SQCM OVERPSI    G/SQ CM    KELVIN     SEC         )
  312 FORMAT(78HOREGION  MATERIAL  NZONES     GAMMA       K-ENERGY
     1 I-ENERGY      TOT ENE       )
  313 FORMAT(I5,2I8,4E15.5)
  314 FORMAT(I6,E15.5)
  315 FORMAT(55H CYCLE   TIME(SEC)    NEXT TIMESTEP  TOTAL ENERGY   JDT)
  316 FORMAT(50H1THE FOLLOWING DATA HAVE BEEN WRITTEN ON TAPE 18- )
  317 FORMAT(53H1                              ***** ENERGY CHECK *****)
  318 FORMAT(I4,1P1E12.5,1P5E10.3,1P1E9.2,1P3E10.3,1P1E9.2,0PF5.2)
  319 FORMAT(116HO J      X                U                 D              E
     1          Q            PQ              TEMP        DTZJ          )
  320 FORMAT(I4,1P8E14.5)
  321 FORMAT(I6)
C
C       GO DIRECTLY TO PRINTING IF TROUBLE HAS OCCURRED.
  324 IF(KUTOFF.NE.0) GO TO 346
C
C     TAPE 18 IS WRITTEN HERE IF-
C       1) NCYCLE=0, OR
C       2) NTAPE  CYCLES HAVE PASSED SINCE LAST WRITING, OR
C       3) NWRITE IS POSITIVE.
C
      IF(NTAPE.LE.0) GO TO 335
  326 NTYMES=NTYMES+1
      IF(NCYCLE.EQ.0) GO TO 329
```

```
C        NWRITE POSITIVE FORCES WRITE TAPE 18 UNLESS NTAPE=0.            OUT1
      IF(NWRITE.GT.0) GO TO 329
C        NTYMES COUNTS THE CYCLES SINCE THE LAST TAPE WRITE.
      IF(NTYMES.LT.NTAPE) GO TO 335
  329 NTYMES=0
      NRECRD=NRECRD+1
      JLAST1=JLAST-1
C
      WRITE(18)      T,(DTMIN(L),L=1,3),NCYCLE,NRECRD,JLAST,JCALC,JCALC1,*
     1 KREZ3B,IMAX,NPROB                                                *
      WRITE(18)      (NZONES(I),INTERJ(I+1),I=1,IMAX),X(JLAST),U(JLAST), *
     1(X(L),U(L),D(L),E(L),Q(L),PQ(L),TEMP(L),DTZJ(L),L=1,JLAST1)       *
      NWRITE=0
      IF(NC(19).LT.1) GO TO 335
      WRITE(6,316)                                                      *
      WRITE(6,330)                                                      *
  330 FORMAT(73H0    T             NCYCLE NRECRD  JLAST   JCALC JCALC1   KRE
     1Z3B IMAX NPROB    )
      WRITE(6,332)   T,NCYCLE,NRECRD,JLAST,JCALC,JCALC1,KREZ3B,IMAX,     *
     1 NPROB                                                            *
  332 FORMAT(1H0,1P1E14.5,8I7)
      WRITE(6,331)   (DTMIN(L),L=1,3)                                   *
  331 FORMAT(10H0DTMIN(L)=1P3E16.7)
      WRITE(6,333)   (NZONES(I),I=1,IMAX)                               *
  333 FORMAT(11H0NZONES(I)=20I5)
      IMAX1=IMAX+1
      WRITE(6,334)   (INTERJ(I),I=1,IMAX1)                              *
  334 FORMAT(11H INTERJ(I)=20I5)
      WRITE(6,319)                                                      *
      WRITE(6,320)   (L,X(L),U(L),D(L),E(L),Q(L),PQ(L),TEMP(L),DTZJ(L), *
     1 L=1,JLAST1)                                                      *
      WRITE(6,320)   JLAST,X(JLAST),U(JLAST)                            *
C
  335 IF(NPRINT.GT.0) GO TO 341
      IF(NCYCLE.GT.5) GO TO 338
  336 MCY=MCY+1
      GO TO 341
C
C     PRINTOUT OCCURS HERE IF EITHER-
C        1) NPR CYCLES WERE CALCULATED SINCE THE LAST PRINTOUT, OR
C        2) SOME SUBROUTINE HAS SET NPRINT=1, OR
C        3) NCYCLE IS LESS THAN 5, OR
C        4) TIME TLIST(7) HAS BEEN REACHED, OR
C        5) TROUBLE HAS OCCURRED (KUTOFF=1).
C
  338 IF(T.GE.TLIST(7)) GO TO 3400
C     MCY COUNTS THE CYCLES SINCE THE LAST PRINT
  339 MCY=MCY+1
      IF(MCY.LT.NPR) RETURN
```

```
 3400 IF(NC(6)) 340,340,3410                                           OUT1
 3410 TLIST(7)=TLIST(7)*TFACTR
      MCY=0
      GO TO 346
  340 MCY=0
  341 NPRINT=0
      OUTDT=TLIST(1)
      IF(TLIST(7).GE.TLIST(2)) OUTDT=TLIST(3)
  345 TLIST(7)=T+OUTDT
C
C       CALC TOTAL INTERNAL AND KINETIC ENERGY IN EACH REGION.
  346 EINSUM=0.
      EKSUM=0.
      ESUM=0.
      DO 355 I=1,IMAX
      EINT(I)=0.
      EKIN(I)=0.
      JMIN=INTERJ(I)
      JMAX=INTERJ(I+1)-1
      DO 350 J=JMIN,JMAX
      JPH=J
      USQ=(U(J)**2 +U(J+1)**2)*.5
      EINT(I)=EINT(I) +E(JPH)*GRAMS(JPH)
      EKIN(I)=EKIN(I) +.5*USQ*GRAMS(JPH)
      PDYN(JPH)=.5*D(JPH)*USQ*14.50382E-6
  350 CONTINUE
      EINSUM=EINSUM+EINT(I)
  355 EKSUM=EKSUM+EKIN(I)
      ETOTAL=EINSUM+EKSUM
      ETENTH=.1*ESTART
C       SUPPRESS ENERGY CHECK IF NC(4)=1
      IF(NC(4).EQ.1) GO TO 360
  357 IF(ABS (ESTART-ETOTAL)-ETENTH) 360,360,358
C       TERMINATE PROBLEM IF TOTAL ENERGY CHANGES TEN PERCENT.
  358 WRITE(6,317)                                                    *
      KUTOFF=1
C
C
  360 WRITE(6,303)    NPROB,(ALIST(L),L=13,24)                        *
  363 WRITE(6,315)                                                    *
      WRITE(6,305)   NCYCLE,T,DTMIN(2),ETOTAL,JDT                     *
C
      JPRINT=JCALC-1
C       NC(17)=1 CAUSES PRINTING OF CUMULATIVE MASSES (GRAMS/SQ CM
C       BETWEEN ORIGIN AND X) IN PLACE OF PDYN.
      IF(NC(17).LE.0) GO TO 3639
      IF(NCYCLE.EQ.0) JPRINT=JLAST-1
      PDYN(1)=D(1)*(X(2)-X(1))
      DO 3631 JPH=2,JPRINT
 3631 PDYN(JPH)=PDYN(JPH-1)+D(JPH)*(X(JPH+1)-X(JPH))
```

```
      GO TO 366                                                        OUT1
C       MAS=1 CAUSES PRINTING OF ZONE MASSES IN PLACE OF PDYN.
 3639 IF(MAS.LE.0) GO TO 369
      JPRINT=JLAST-1
  364 DO 365 JPH=1,JPRINT
  365 PDYN(JPH)=GRAMS(JPH)
  366 WRITE(6,306)                                                     *
      WRITE(6,307)                                                     *
      MAS=0
      GO TO 370
C
  369 WRITE(6,308)                                                     *
      WRITE(6,309)                                                     *
  370 IF(NCYCLE.EQ.0) JPRINT=JLAST-1
      DO 361 J=1,JPRINT
      XAV(J)=(X(J)+X(J+1))*.5
  361 CONTINUE
      DO 371 L=1,JPRINT
  371 PSI(L)=PQ(L)*14.50382E-6
      IF(NC(11)) 377,377,3710
C       CALC OVERPRESSURES IF NC(11)=1 (3710 THRU 375).
 3710 DO 375 L=1,JPRINT
      IF(IARDC.GE.31) GO TO 372
      ALTCM=COSPHI*(X(L)+X(L-1))*.5 +HOBKM*1.E5
          CALL ARDC(ALTCM,PRATIO,TRATIO,DRATIO)
      PRAMB=PRATIO*14.696
      GO TO 373
  372 PRAMB=PQ(JLAST-1)*14.50382E-6
  373 PSI(L)=PSI(L)-PRAMB
      IF(ABS(PSI(L))*1.E+5.LT.PRAMB) PSI(L)=0.
  375 CONTINUE
  377 WRITE(6,318)   NOUGHT,X(1),U(1)                                  *
      JSTART=1
      DO 385 II=1,IMAX
      JQUIT=JSTART+NZONES(II)-1
      IF(JQUIT.LT.JPRINT) GO TO 380
      IF(JSTART.GE.JPRINT) GO TO 386
  379 JQUIT=JPRINT
  380 WRITE(6,318)   (L,X(L+1),U(L+1),XAV(L),D(L),E(L),PQ(L),Q(L),     *
     1 PSI(L),PDYN(L),TEMP(L),DTZJ(L),GAMMAJ(L),L=JSTART,JQUIT)        *
      WRITE(6,381)                                                     *
  381 FORMAT(1H )
      JSTART=JQUIT+1
  385 CONTINUE
  386 WRITE(6,312)                                                     *
      DO 390 I=1,IMAX
      ENTOT(I)=EKIN(I)+EINT(I)
  390 CONTINUE
  391 WRITE(6,313)   (I,NEQST(I),NZONES(I),GAMMAI(I),EKIN(I),EINT(I),  *
     1 ENTOT(I),I=1,IMAX)                                              *
C
```

```
C        LOCATE ZONE JQMAX WHICH CONTAINS MAX Q.                        OUT1
      QMAX=0.
      JQMAX=0
      DO 393 J=1,JCALC
      IF(Q(J).LE.QMAX) GO TO 393
  392 QMAX=Q(J)
      JQMAX=J
  393 CONTINUE
      IF(JQMAX.EQ.0) GO TO 399
  394 IF(JQMAX.GE.JLAST-2) GO TO 399
C        INTERPOLATE WITHIN ZONE JQMAX FOR BETTER Q LOCATION.
  396 QFRAC=(Q(JQMAX)-Q(JQMAX-1))/(Q(JQMAX)-Q(JQMAX+1)+1.E-15)
      XQMAX=(X(JQMAX)+X(JQMAX+1)*QFRAC)/(QFRAC+1.)
  397 WRITE(6,398)    JQMAX,XQMAX                                          *
  398 FORMAT(23H0MAXIMUM Q    IS IN ZONE,I4,6H AT X=,1PE11.4)
C        FIND MAXIMUM PRESSURE.
      JPMAX=JQMAX
      DO 405 J=1,JCALC
      IF(PQ(J    ).GT.PQ(JPMAX)) JPMAX=J
  405 CONTINUE
 4050 XPMAX=XAV(JPMAX)
      WRITE(6,406) JPMAX,XPMAX                                             *
  406 FORMAT(23H MAXIMUM PSI IS IN ZONE,I4,6H AT X=,1PE11.4)
C        STOP PROBLEM IF TROUBLE HAS OCCURRED.
  399 IF(KUTOFF.EQ.0) RETURN
      IF(NC(12).GT.0) CALL  DUMP(AINQ(1),ZONSIZ,1,AINQ(1),ZONSIZ,2)
      STOP
       END
```

```
$IBFIC OUT2    LIST,REF
      SUBROUTINE OUT2                                                   OUT2
C
C     USUALLY, KOUT2=4, KOUT2A=10, KOUT2B=0, NC(9)=0,NC(18)=1.
C     KOUT2  = 0   THIS PROGRAM NOT USED.
C     KOUT2  =+1   OMIT EXTRAPOLATION OF PRESSURE.
C     KOUT2  =+2   INCLUDE EXTRAPOLATED PQ IN OUTPUT.
C     KOUT2  GR+2  BASE THE EXTRAPOLATION ON KOUT2 POINTS PER GROUP.
C     KOUT2A =     PREPARE A LINE OF OUTPUT EVERY KOUT2A-TH CYCLE.
C     KOUT2B = L   USE S(1 THRU L) AS OVERPRESSURES AND FIND THE DIST-
C                  ANCES AT WHICH THEY OCCUR.  WHEN A PRESSURE S(N) IS
C                  REACHED, REPLACE S(N) BY THE DISTANCE AND INCREASE
C                  NC(26) BY 1.  (THESE DISTANCES ARE BEING FOUND FOR
C                  USE BY OUT3).
C     KOUT2B = 0   OMIT THE S(L) SECTION.
C     NC(9)       =0  PUT OUT2 OUTPUT ON TAPE 6.
C                 =2  PUT OUT2 OUTPUT ON TAPE 26.
C     NC(18)          DO NOT CARRY MAX Q SEARCH INWARD INTO REGION NC(18)
C
      DIMENSION  MSY(50),TAIM(50),MJ(50),XPMAX(50),OVPQ(300),JBACK(50)  OUT2
     1         ,OVPQEX(50),AMBPSI(50),DLPDR(50),XQMAX(50),OVPMAX(50)    OUT2
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
      DATA NUFSED/0/
C
      IF(IDENT.NE.1) GO TO 7100
  701 WRITE(6,702)                                                      *
  702 FORMAT(50HOSUBROUTINE OUT2 VERSION  3,  3- 1-65   TAPE 6,26   )
 7100 IF(NUFSED.EQ.1234) GO TO 710
      NUFSED=1234
      XMAXQ=0.
      QHUGE=1.
      ICOUNT=0
      MOO=0
      NPTS=4
      NC(26)=0
      IF(KOUT2-2) 710,704,705
  704 NILE=4
      GO TO 706
  705 NILE=KOUT2
  706 NILE2=2*NILE
      SNILE=NILE
      RETURN
C
  710 ICOUNT=ICOUNT+1
      IF(NQUIT.LT.0) GO TO 716
  715 IF(ICOUNT.LT.KOUT2A) RETURN
  716 ICOUNT=0
C
```

```
C        LOCATE ZONE JQMAX WITH HIGHEST Q.                               OUT2
C
      QMAX=0.
      JQMAX=1
      NC18=NC(18)
      JBEGIN=INTERJ(NC18+1)
      JCALC1=JCALC-1
      DO 721 J=JBEGIN,JCALC1
      IF(Q(J).LE.QMAX) GO TO 721
  720 QMAX=Q(J)
      JQMAX=J
  721 CONTINUE
C        INTERPOLATE WITHIN ZONE JQMAX FOR BETTER Q LOCATION.
C        USE MIDDLE OF ZONE IF JQMAX IS AT EDGE OF PROBLEM.
      IF(JQMAX.LE.1) GO TO 723
  722 IF(JQMAX.LT.JLAST-1) GO TO 725
  723 XMAXQ=(X(JQMAX+1)+X(JQMAX))*.5
      GO TO 726
  725 QFRAC=(Q(JQMAX)-Q(JQMAX-1))/(Q(JQMAX)-Q(JQMAX+1)+1.E-15)
      XQWAS=XMAXQ
      XMAXQ=(X(JQMAX)+X(JQMAX+1)*QFRAC)/(QFRAC+1.)
      IF(XMAXQ.GE.XQWAS) GO TO 726
      XMAXQ=XQWAS
      GO TO 732
  726 MOO=MOO+1
      XQMAX(MOO)=XMAXQ
      MSY(MOO)=NCYCLE
      TAIM(MOO)=T
      MJ(MOO)=JQMAX
C        CALCULATE AMBIENT PRESSURE AT LOCATION OF MAX Q.
      IF(IARDC-31) 729,728,728
  728 AMBPSI(MOO)=PQ(JCALC-1)*14.50382E-6
      GO TO 7301
  729 ALTCM=COSPHI*XQMAX(MOO)+HOBKM*1.E5
          CALL APDC(ALTCM,PRATIO,TRATIO,DRATIO)
      AMBPSI(MOO)=14.696*PRATIO
C
C        SEARCH FOR LARGEST OVERPRESSURE.
C
C        CALCULATE OVERPRESSURES. (THRU 7305).
 7301 DO 7305 L=JBEGIN,JCALC1
      IF(IARDC-31) 7303,7302,7302
 7302 AMBPR=PQ(JCALC-1)*14.50382E-6
      GO TO 7304
 7303 ALTCM=COSPHI*(X(L+1)+X(L))*.5+HOBKM*1.E5
          CALL ARDC(ALTCM,PRATIO,TRATIO,DRATIO)
      AMBPR=14.696*PRATIO
 7304 OVPQ(L)=PQ(L)*14.50382E-6 -AMBPR
 7305 CONTINUE
C
```

```
  730 PQBIG=OVPQ(JBEGIN)                                                OUT2
      JPQMAX=JBEGIN
      DO 735 J=1,JCALC1
      IF(OVPQ(J).LE.PQBIG) GO TO 735
      PQBIG=OVPQ(J)
      JPQMAX=J
  735 CONTINUE
      JSAVE=JQMAX-JPQMAX
C        MAXIMUM OVERPRESSURE WAS FOUND IN ZONE JPQMAX.
C        JSAVE IS NUMBER OF ZONES BACK FROM MAX Q TO MAX OVPQ.
      IF(JPQMAX.LE.JBEGIN) GO TO 732
      IF(JSAVE.LT.24) GO TO 739
  732 XPMAX(MOO)=-0.
      OVPMAX(MOO) = -0.
      OVPQEX(MOO)=-0.
      JBACK(MOO)=0
      GO TO 7005
  739 XPMAX(MOO)=(X(JPQMAX)+X(JPQMAX+1))*.5
C
  740 OVPMAX(MOO) = PQBIG
      JBACK(MOO)=JSAVE
C
C        TEST IF EXTRAPOLATION DESIRED
C
      IF(KOUT2.LT.2) GO TO 753
C
C        CALCULATE EXTRAPOLATED OVERPRESSURE BY EXTRAPOLATING P VS X DATA
C        TO THE POSITION OF MAXIMUM Q.
  741 IF(JPQMAX-JBEGIN.GT.NILE2) GO TO 743
  742 OVPQEX(MOO) = -0.
      GO TO 753
  743 SUMA=0.
      SUMB=0.
      DO 745 J=1,NILE
      JJ=JPQMAX+1-J
      SUMA=SUMA+OVPQ(JJ)
      SUMB=SUMB+X(JJ)
  745 CONTINUE
      SUMC=0.
      SUMD=0.
      DO 748 J=1,NILE
      JJ=JPQMAX-NILE+1-J
      SUMC=SUMC+OVPQ(JJ)
      SUMD=SUMD+X(JJ)
  748 CONTINUE
      PAV1=SUMA/SNILE
      XAV1=SUMB/SNILE
      PAV2=SUMC/SNILE
      XAV2=SUMD/SNILE
      SLOPE=(PAV1-PAV2)/(XAV1-XAV2)
```

```
      TERCEP=PAV2-SLOPE*XAV2                                            OUT2
      OVPQEX(MOO)=SLOPE*XQMAX(MOO)+TERCEP
C
C       SPECIAL OPTION FOR TRIGGERING OUT3 AT DESIRED PRESSURE LEVELS.
C       KOUT2B=NUMBER OF PRESSURE LEVELS.  READ IN ON S CARDS.
  753 IF(KOUT2B) 7005,7005,7000
 7000 NLOC=NC(26) +1
      IF(OVPQEX(MOO)) 7005,7005,7001
 7001 IF(OVPQEX(MOO).GT.S(NLOC)) GO TO 7005
      S(NLOC+12)=AMBPSI(MOO)                                                  C
 7002 S(NLOC)=XQMAX(MOO)
      NC(26)=NLOC
      IF(KOUT2B.LE.NLOC) KOUT2B=0
C
C       PRINT WHEN 50 LINES OF DATA HAVE BEEN STORED OR IF NQUIT IS
C       NEGATIVE (INDICATING COMPLETION OF PROGRAM)
C
 7005 IF(NQUIT.LT.0) GO TO 761
  756 IF(MOO.LT.50) RETURN
  761 IF(NC(9).GT.0) GO TO 7610
      WRITE( 6,762)                                                     *
      WRITE( 6,7620)                                                    *
      WRITE( 6,763)
      GO TO 764
 7610 WRITE(26,762)                                                     *
      WRITE(26,7620)                                                    *
      WRITE(26,763)                                                     *
  762 FORMAT(17H1SHOCK FRONT DATA    )
 7620 FORMAT(1H0,31X,42HEXTRAPOLATED VALUES   DIRECTLY READ VALUES     )
  763 FORMAT(119H  CYCLE     J  JBACK      TIME    X(MAX Q)     MAX OVP    X(
     1MAX OVP)    MAX OVP  SLOPE         AMBPSI                              )
C       MOO=NUMBER OF LINES OF DATA STORED.
  764 DO 790 J=1,MOO
      IF(J.LT.5) GO TO 781
      IF(J.GT.MOO-5) GO TO 781
C
C     CALCULATE LOGARITHMIC SLOPE OF OVERPRESSURE VS DISTANCE CURVE
C     USING 4 POINTS ON EACH SIDE OF DESIRED POINT.
C
  766 PSUM1=0.
      PSUM2=0.
      XSUM1=0.
      XSUM2=0.
      DO 780 JJ=1,4
      JM=J-JJ
      JP=J+JJ
      IF (OVPMAX(JM) .LE. 0.) GO TO 781
      IF (OVPMAX(JP) .LE. 0.) GO TO 781
  779 PSUM1 = PSUM1 + ALOG10(OVPMAX(JM))
      PSUM2 = PSUM2 + ALOG10(OVPMAX(JP))
```

OUT2

```
      XSUM1=XSUM1+ALOG10(XQMAX(JM))
      XSUM2=XSUM2+ALOG10(XQMAX(JP))
  780 CONTINUE
      DLOGP=PSUM1-PSUM2
      DLOGX=XSUM1-XSUM2
      DLPDX=DLOGP/DLOGX
      GO TO 782
  781 DLPDX=0.
  782 DLPDR(J)=DLPDX
C
      IF(NC(9).GT.0) GO TO 788
      WRITE( 6,789) MSY(J),MJ(J),JBACK(J),TAIM(J),XQMAX(J),OVPQEX(J),     *
     1 XPMAX(J),OVPMAX(J),DLPDR(J),AMBPSI(J)                               *
      GO TO 790
  788 WRITE(26,789) MSY(J),MJ(J),JBACK(J),TAIM(J),XQMAX(J),OVPQEX(J),     *
     1 XPMAX(J),OVPMAX(J),DLPDR(J),AMBPSI(J)                               *
  789 FORMAT(1H ,I6,I5,I4,1P2E12.5,1PE10.3,1PE14.5,1PE10.3,0PF7.3,
     1 1PE14.5)
  790 CONTINUE
      MOO=0
  799 RETURN
      END
```

```
$IBFTC OUT3    LIST,REF
      SUBROUTINE OUT3                                                   OUT3
C
C     THIS SUBROUTINE PREPARES SUMMARY SHEETS OF P, U, DENSITY, AND
C       DYNAMIC PRESSURE VERSUS TIME AT UP TO 8 DISTANCES, S(1-8).
C       IF KOUT3=0, THIS SUBROUTINE IS NOT CALLED.
C       USUALLY, NC(10)=0, NC(15)=0, KOUT3=0 OR 4, KOUT3A=10 .
C
C     NC(10)      =0  PUT OUT3 OUTPUT ON TAPE 6.
C                 =2  PUT OUT3 OUTPUT ON TAPE 26.
C     NC(15)      =WHEN SHOCK ARRIVES, STORE DATA EVERY CYCLE INSTEAD OF
C                  EVERY KOUT3A CYCLES FOR NC(15) CYCLES.  USE 150 IF 0 .
C     KOUT3 = NUMBER OF POINTS AT WHICH P VS T IS DESIRED (FROM 0 TO 8)
C     KOUT3A= STORE A LINE OF DATA EVERY KOUT3A-TH CYCLE.
C
C     IF KOUT2B IS NEGATIVE,
C     THIS SUBROUTINE COMPUTES THE MOTION (RADIAL DISTANCE FROM BURST
C     CENTER VS TIME) OF THE POINTS AT WHICH PT DATA ARE BEING TAKEN.
C     AT THE BEGINNING OF THE PROBLEM, THIS SUBROUTINE EXPECTS TO READ
C     DATA CARDS FOR RA, RB, AND SPEED.  THESE CARDS ARE TO BE PLACED
C     AFTER THE REGULAR INPUT.
C
C     RY = DISTANCE (CM) TO NEAREST POINT ON (STRAIGHT-LINE) TRAJECTORY.
C     RX = DISTANCE (CM) ALONG TRAJECTORY MEASURED FROM NEAREST-TO-BURST
C          POINT IN DIRECTION OF MOTION.  (I.E., ONCOMING OBJECT HAS
C          NEGATIVE RX, OBJECT GOING AWAY HAS POSITIVE RX.)
C     SPEED = SPEED(CM/SEC) OF OBJECT ALONG TRAJECTORY.
C
      DIMENSION RY(8),RX(8),SPEED(8),RXINIT(8),DIREC(160),RL(160)       OUT3
     1,PQPSI(160),DENSTY(160),TAIM(160),R(8),MM(8),NCYKLE(160)          OUT3
     2,AMBPSI(8),OVPSI(160),UPT(160),MSTORE(8),MCOUNT(8),NUFL(8)        OUT3
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
      DATA NONCE/0/
C
      IF(IDENT.NE.1) GO TO 804
  801 WRITE(6,802)                                                     *
  802 FORMAT(55HOSUBROUTINE OUT3 VERSION  3,  3- 1-65  TAPE 6,26       )
      RETURN
C
C       THIS SECTION (804 THRU 8) IS USED ONLY ONCE.
  804 IF(NONCE-1234) 805,807,805
  805 DO 806 LOC=1,KOUT3
      MM(LOC)=0
      NUPH=150
      IF(NC(15).GT.0) NUPH=NC(15)
      NUFL(LOC)=NUPH
      MSTORE(LOC)=KOUT3A
```

```
      MCOUNT(LOC)=0                                                     OUT3
  806 CONTINUE
      NLINES=160/KOUT3
      IF(NLINES.GT.50) NLINES=50
      NONCE=1234
      KPT=KOUT3
      NPT=KOUT3
      IF(KOUT2B) 1,9,899
    9 IF(IARDC.GT.30) GO TO 6
      DO 10 L=1,KOUT3
      R(L)=S(L)
      ALTCM=COSPHI*R(L)+HOBKM*1.E5
          CALL ARDC(ALTCM,PRATIO,TRATIO,DRATIO)
      AMBPSI(L)=14.696*PRATIO
   10 CONTINUE
      GO TO 807
    1 READ(5,2)   (RY(L),L=1,NPT)                                      *
      READ(5,2)   (RXINIT(L),L=1,NPT)                                  *
      READ(5,2)   (SPEED(L),L=1,NPT)                                   *
    2 FORMAT(1P6E10.5)
      IF(NC(10).NE.2) GO TO 7
      WRITE(26,3)   (RY(L),L=1,NPT)                                    *
      WRITE(26,4)   (RXINIT(L),L=1,NPT)                                *
      WRITE(26,5)   (SPEED(L),L=1,NPT)                                 *
      GO TO 6
    7 WRITE(6,3)    (RY(L),L=1,NPT)                                    *
      WRITE(6,4)    (RXINIT(L),L=1,NPT)                                *
      WRITE(6,5)    (SPEED(L),L=1,NPT)                                 *
    3 FORMAT(7H RY    =1P8E14.5)
    4 FORMAT(7H RX    =1P8E14.5)
    5 FORMAT(7H SPEED=1P8E14.5)
    6 DO 8 L=1,8
      AMBPSI(L)=PQ(JCALC -1)*14.50382E-6
      R(L)=S(L)
    8 CONTINUE
      GO TO 807
C
C       DATA IS STORED FOR LOCATION LOC EVERY MSTORE(LOC) CYCLES.
C       MCOUNT(LOC)=NUMBER OF CYCLES SINCE LAST PRINT FOR LOCATION LOC.
  807 IF(KOUT2B) 812,812,808
C       BYPASS OUT3 UNTIL OUT2 HAS SET NC(26) NON-ZERO.
  808 KPT=NC(26)
      IF(KPT.LE.0) RETURN
  810 IF(NUFL(LOC).EQ.NUPH) MSTORE(LOC)=1
  812 DO 890 LOC=1,KPT
      MCOUNT(LOC)=MCOUNT(LOC)+1
      IF(NQUIT.LT.0) GO TO 815
      IF(NCYCLE.LT.5) GO TO 890
  814 IF(MSTORE(LOC).GT.MCOUNT(LOC)) GO TO 890
  815 MCOUNT(LOC)=0
C       SEEK DATA TO PUT IN LINE KK.
      KK=MM(LOC)+1
```

```
      KKSLOC=NLINES*(LOC-1)+KK                                          OUT3
      IF(KOUT2B) 816,833,832
  816 RX(LOC)=RXINIT(LOC)+SPEED(LOC)*T
      RRR=SQRT (RY(LOC)**2+RX(LOC)**2)
      R(LOC)=AMIN1(RRR,X(JLAST-1))
      GO TO 833
  832 R(LOC)=S(LOC)
      AMBPSI(LOC)=S(LOC+12)
C
C       FIND ZONE JJJ IN WHICH DESIRED R LIES.
C
  833 JJJ=JLAST-1
      DO 835 L=2,JLAST
      IF(X(L).LT.R(LOC)) GO TO 835
  834 JJJ=L-1
      GO TO 836
  835 CONTINUE
C
  836 PHRAC=(R(LOC)-X(JJJ))/(X(JJJ+1)-X(JJJ))
      UPT(KKSLOC)=U(JJJ) + PHRAC*(U(JJJ+1)-U(JJJ))
C       NABOR IS THE NEARER OF THE TWO ZONES ADJACENT TO JJJ.
      IF(PHRAC.LT.0.5) GO TO 838
  837 NABOR=JJJ+1
      GO TO 839
C       USE ZONE 2 AS NABOR IF JJJ=1 .
  838 IF(JJJ.LE.1) GO TO 837
 8381 NABOR=JJJ-1
  839 CTR=(X(JJJ+1)+X(JJJ))/2.
      CTRADJ=(X(NABOR)+X(NABOR+1))/2.
C       FRAC IS INTERPOL FACTOR FOR QUANTITIES DEFINED AT ZONE CENTERS
      FRAC=(R(LOC)-CTR)/(CTRADJ-CTR)
      PQPSI(KKSLOC)=(PQ(JJJ)+FRAC*(PQ(NABOR)-PQ(JJJ)))*14.50382E-6
      RL(KKSLOC)=R(LOC)
      IF(KOUT2B) 8390,8400,8400
C
C       CALC NET PARTICLE VELOCITY INCLUDING THAT DUE TO STA MOTION.
 8390 UTRAJ=SPEED(LOC)-UPT(KKSLOC)*RX(LOC)/R(LOC)
      UNORM=UPT(KKSLOC)*RY(LOC)/R(LOC)
      UPT(KKSLOC)=SQRT (UTRAJ**2+UNORM**2)
C       CALC DIRECTION COSINE OF FLOW WITH RESPECT TO TRAJECTORY.
C       DIREC IS COSINE OF ANGLE BETWEEN STATION DIRECTION OF MOTION AND
C       DIRECTION OF PARTICLE MOTION.
C       DIREC=1 IS HEAD-ON,  DIREC=0 IS SIDE-ON,  DIREC=-1 IS TAIL-ON.
      DIREC(KKSLOC)=UTRAJ/UPT(KKSLOC)
C
C     NOTE THAT THE TRAJECTORY IN THIS FORMULATION REMAINS A STRAIGHT
C     LINE THROUGHOUT THE PROBLEM.  IF DESIRED, THE TRAJECTORY COULD BE
C     MADE TO VARY ACCORDING TO CALCULATED FORCES ON THE STATION.
C
 8400 DENJJJ=D(JJJ)
```

```
      DENNAB=D(NABOR)                                                 OUT3
      DENSTY(KKSLOC)=DENJJJ+FRAC*(DENNAB-DENJJJ)
      TAIM(KKSLOC)=T
      OVPSI(KKSLOC)=PQPSI(KKSLOC)-AMBPSI(LOC)
      IF(ABS (OVPSI(KKSLOC))-.0001*AMBPSI(LOC)) 840,840,841
  840 OVPSI(KKSLOC)=0.
  841 NCYKLE(KKSLOC)=NCYCLE
C
C     PRINT WHEN NLINES LINES OF DATA ARE STORED FOR A GIVEN R.
C     NQUIT NEGATIVE MEANS PROBLEM IS DONE AND EVERYTHING IS PRINTED.
C     MM(LOC) IS THE NUMBER OF LINES OF DATA STORED SO FAR IN LIST LOC.
C
      MM(LOC)=MM(LOC)+1
C       NUFL(LOC) IS NUPH UNTIL THE SHOCK ARRIVES.
      NUF=NUFL(LOC)
      IF(NUF) 846,8450,842
  842 IF(NUF.LT.NUPH) GO TO 845
  843 IF(OVPSI(KKSLOC)) 889,889,844
  844 MSTORE(LOC)=1
  845 NUFL(LOC)=NUFL(LOC)-1
      IF(KOUT2B) 846,8451,8451
 8451 RY(LOC)=S(LOC)
      GO TO 846
 8450 MSTORE(LOC)=KOUT3A
      NUFL(LOC)=-1
C
  846 IF(NQUIT) 848,847,847
  847 IF(KK.LT.NLINES) GO TO 890
C       PRINTING IS FORCED WHEN 848 IS REACHED.
  848 IF(NC(10).NE.2) GO TO 850
 8500 WRITE(26,851)   LOC                                          *
      WRITE(26,852)   AMBPSI(LOC)                                  *
      WRITE(26,8521)  SPEED(LOC)                                   *
      WRITE(26,8522)  RY(LOC)                                      *
      WRITE(26,853)                                                *
      GO TO 8530
  850 WRITE( 6,851)   LOC                                          *
      WRITE( 6,852)   AMBPSI(LOC)                                  *
      WRITE( 6,8521)  SPEED(LOC)                                   *
      WRITE( 6,8522)  RY(LOC)                                      *
      WRITE( 6,853)                                                *
  851 FORMAT(10H1  STATION,I2)
  852 FORMAT(25H    AMBIENT PRESSURE(PSI)=,1PE12.5)
 8521 FORMAT(25H    SPEED(CM/SEC)        =,1PE12.5)
 8522 FORMAT(25H    PERI-BURST DIST(CM)  =,1PE12.5)
  853 FORMAT(106H0  CYCLE     TIME         PSI         OVPSI        DYN PSI
     1     DENSTY       PART VEL   DIREC COS    RADIAL R    )
 8530 DO 860 LM=1,KK
      LMSLOC=NLINES*(LOC-1)+LM
      DYNPSI=.5*DENSTY(LMSLOC)*UPT(LMSLOC)**2*14.50382E-6
```

```
      IF(NC(10).NE.2) GO TO 8540                                        OUT3
 8541 WRITE(26,854)  NCYKLE(LMSLOC),TAIM(LMSLOC),PQPSI(LMSLOC), OVPSI(LM*
     1SLOC),DYNPSI,DENSTY(LMSLOC),UPT(LMSLOC),DIREC(LMSLOC),RL(LMSLOC)   *
      GO TO 860
 8540 WRITE( 6,854)  NCYKLE(LMSLOC),TAIM(LMSLOC),PQPSI(LMSLOC), OVPSI(LM*
     1SLOC),DYNPSI,DENSTY(LMSLOC),UPT(LMSLOC),DIREC(LMSLOC),RL(LMSLOC)   *
  854 FORMAT(1H ,I6,1P8E12.3)
  860 CONTINUE
  889 MM(LOC)=0
C        END PRINTING.
  890 CONTINUE
C        END STATION (LOC) LOOP.
  899 RETURN
      END
```

```
$IBFTC OUT4    LIST,REF
      SUBROUTINE OUT4
C     PUNCH CARDS WHEN CYCLE KOUT4 IS REACHED.                          OUT4
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
      IF(IDENT.NE.1) GO TO 90
      WRITE(6,89)
   89 FORMAT(50H0SUBROUTINE OUT4 VERSION  3,  3- 1-65  PUNCH CARDS)
      RETURN
C
   90 IF(NCYCLE.NE.KOUT4) RETURN
      KOUT4=0
      INCODS=1
      PUNCH 1,     (ALIST(L),L=1,12)                                    *
      PUNCH 1,     (ALIST(L),L=13,24)                                   *
    1 FORMAT(12A6)
      PUNCH 2,     NPROB ,K     ,MURIN ,MUREX ,IMAX  ,IARDC ,           *
     1             INTAPE,INCODS,NQUIT ,NPR   ,NTAPE ,JCALC             *
      PUNCH 2,     KOUT2 ,KOUT2A,KOUT2B,KOUT3 ,KOUT3A,KOUT4 ,           *
     1             KREZ1 ,KREZ2 ,KREZ3 ,KREZ3A,KREZ3B,KREZ4             *
      PUNCH 2,     (NC(L),L=1,12)                                       *
      PUNCH 2,     (NC(L),L=13,24)                                      *
    2 FORMAT(12I5)
      PUNCH 3,     (I ,NEQST(I),GAMMAI(I),NZONES(I),OUTBDY(I),EINIT(I), *
     1             UINIT(I),DINIT(I),CINQ(I),AINQ(I),ZONING(I),I=1,IMAX)*
    3 FORMAT(I2,I3,0PF6.3,I4,1P4E10.3,0P3F5.2)
      PUNCH 4,     COSPHI,HOBKM,WKT   ,BLANK1,BLANK2,ZONSIZ             *
      PUNCH 4,     T,TREZO,DTMIN(2),DTRATE,STABIL,UCUT                  *
      PUNCH 4,     (TLIST(I),I=1,6)                                     *
      PUNCH 4,     (S(L),L=1,12)                                        *
    4 FORMAT(1P6E10.3)
      PUNCH 5,NCYCLE
    5 FORMAT(15H $LIST1 NCYCLE=,I5,10H,INCODS=2$)
      ZERO=0.
      JLAST1=JLAST-1
      PUNCH 91,    (L,X(L),U(L),D(L),E(L),Q(L),L=1,JLAST1)              *
      PUNCH 91,    JLAST,X(JLAST),U(JLAST),ZERO,ZERO,ZERO               *
      N300=300
      PUNCH 91,    N300,ZERO,ZERO,ZERO,ZERO,ZERO                        *
   91 FORMAT(I4,1P4E14.7,1P1E12.5)
      NULL=0
      PUNCH 2,     NULL                                                 *
      RETURN
       END
```

```
$IBFTC REZ1    LIST,REF
      SUBROUTINE REZ1                                                   REZ1
C                                                 REMOVE SQUASHED ZONE
C
C     THIS SUBROUTINE CUTS RUNNING TIME BY COMBINING THE ZONE WITH THE
C       SMALLEST TIME STEP WITH THE ADJACENT INNER ZONE, UNLESS THIS
C       WOULD RESULT IN THE REMOVAL OF AN INTERFACE, IN WHICH CASE THE
C       ZONE IS COMBINED WITH THE ADJACENT OUTER ZONE.  REZONING WILL
C       NOT OCCUR IN REGIONS OF LESS THAN 6 ZONES OR IN REGIONS BEYOND
C       REGION KREZ1.  KREZ1 SHOULD BE SET SMALL ENOUGH SO NONE OF THE
C       FINE ZONES OF REZ3 ARE INVOLVED.  (REMOVING ONE OF THE FINE
C       ZONES WOULD FOUL UP REZ3).
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
      IF(IDENT.NE.1) GO TO 403
  401 WRITE(6,402)                                                    *
  402 FORMAT(55H0SUBROUTINE REZ1 VERSION  3,  3- 1-65  TAPES 6, 26      )
      TBIG=0.
      RETURN
C
C     TBIG IS LARGEST TIMESTEP USED THUS FAR IN PROBLEM.
  403 IF(KREZ1.EQ.1) GO TO 404
      IF(NCYCLE.EQ.KREZ1) GO TO 412
  404 IF(DTMIN(3).LE.TBIG) GO TO 406
      TBIG=DTMIN(3)
      RETURN
  406 IF(DTMIN(3).GT.0.7*TBIG) RETURN
C
C     ZONE JDT HAS SMALLEST TIMESTEP
C     DETERMINE REGION THAT CONTAINS ZONE JDT.
C
  412 NJ=JDT
      DO 420 I=1,IMAX
      IF(NJ-INTERJ(I+1)) 418,419,420
  418 NJREG=I
      GO TO 425
  419 NJREG=I+1
      GO TO 425
  420 CONTINUE
C
C      REGION(NJREG) CONTAINS ZONE(NJ),IF REGION(NJREG)CONTAINS
C       LESS THAN SIX ZONES,REZONING IS NOT CARRIED OUT
C
  425 IF(NZONES(NJREG).LT.6) RETURN
C
C       IF NJREG EXCEEDS KREZ1, REZONING IS NOT CARRIED OUT.
C
      IF(NJREG.GT.KREZ1) RETURN
C
```

```
C     TEST TO DETERMINE WHETHER ZONE(NJ) IS BOUNDED ABOVE BY A REGION   REZ1
C       INTERFACE,IF IT IS SET NJ=NJ+1
C
  426 DO 427 I=1,IMAX
      IF(NJ.EQ.INTERJ(I)) GO TO 428
  427 CONTINUE
      GO TO 429
      NJ=NJ+1
C
  429 NPRINT=1
      MAS=1
          CALL OUT1
      L=NJ-1
      IF(NC(14).EQ.0) GO TO 4295
      WRITE(26,430) L,NJ,NJREG                                            *
C       WRITE TAPE 6 ALSO.
 4295 WRITE(6,430) L,NJ,NJREG                                             *
  430 FORMAT(33H0  REZ1 IS ABOUT TO COMBINE ZONES,I4,4H AND,I4,
     1 10H IN REGION,I3)
C
C       COMBINE ZONE(NJ) WITH ZONE(NJ-1)   NOTE THAT L+1=NJ
C       WEIGHT D, E, AND P ACCORDING TO MASS
C
      DMT=GRAMS(L)+GRAMS(L+1)
      X(NJ)=X(NJ+1)
      J=NJ
      NGEOM=2
          CALL GEOM
      D(L)=DMT/VOL
      E(L)=(E(L)*GRAMS(L)+E(L+1)*GRAMS(L+1))/DMT
      P(L)=(P(L)*GRAMS(L)+P(L+1)*GRAMS(L+1))/DMT
      Q(L)=(Q(L)+Q(L+1))*.5
      GRAMS(L)=DMT
      PQ(L)=P(L)+Q(L)
C
C     RELABEL ZONES BEYOND ZONE NJ.
C
      JLAST2=JLAST-2
      DO 440 L=NJ,JLAST2
      X(L)=X(L+1)
      U(L)=U(L+1)
      D(L)=D(L+1)
      E(L)=E(L+1)
      Q(L)=Q(L+1)
      GRAMS(L)=GRAMS(L+1)
C       PQ,DUDT,DTZJ,TEMP,GAMMAJ WILL BE RECALCULATED ANYWAY IN MOTION
C       AND ARE ADJUSTED HERE ONLY TO MAKE THE NEXT PRINTOUT CONSISTENT.
      PQ(L)=PQ(L+1)
      DUDT(L)=DUDT(L+1)
      DTZJ(L)=DTZJ(L+1)
```

```
      TEMP(L)=TEMP(L+1)                                              REZ1
      GAMMAJ(L)=GAMMAJ(L+1)
  440 CONTINUE
      X(JLAST1)=X(JLAST)
      U(JLAST1)=U(JLAST)
      DUDT(JLAST1)=DUDT(JLAST)
C
C
C       CALC INTERJ AND SET SOME CONSTANTS
      NZONES(NJREG)=NZONES(NJREG)-1
      DO 445 I=NJREG,IMAX
      INTERJ(I+1)=INTERJ(I+1)-1
  445 CONTINUE
      JLAST=JLAST1
      JLAST1=JLAST2
      JCALC=JCALC-1
      JCALC1=JCALC-1
      NPRINT=1
      MAS=1
          CALL OUT1
      IF(KREZ3B-NJ) 499,490,490
  490 KREZ3B=KREZ3B-1
  499 RETURN
      END
```

```
$IBFTC REZ2    LIST,REF                                              REZ2
      SUBROUTINE REZ2
C
C     THIS SUBROUTINE DOUBLES THE RANGE OF X, WHILE KEEPING THE NUMBER
C       OF ZONES IN THE PROBLEM THE SAME.  PAIRS OF ADJACENT ZONES
C       ARE AVERAGED AND REPLACED BY ONE LARGER ZONE.
C     IF REZ2 IS USED, THE MAXIMUM NUMBER OF ZONES PERMITTED IN THE
C       PROBLEM IS 299 RATHER THAN 300.
C     FIRST KREZ2-1 REGIONS ARE NOT SUBJECT TO REZONING.
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
      IF(IDENT.NE.1) GO TO 510                                         *
  501 WRITE(6,502)
  502 FORMAT(40H0SUBROUTINE REZ2 VERSION  3, 12-10-64    )
      RETURN
C
  510 JLAST4=JLAST-4
      IF(ABS(U(JLAST4)).LT.UCUT) RETURN
  515 NPRINT=1
      MAS=1
          CALL OUT1                                                    *
      WRITE(6,516)
  516 FORMAT(35H0REZONING BY REZ2 IS ABOUT TO OCCUR)
C
      L2=0
      ISTART=1
      NIX=KREZ2-1
C     DO NOT DO ANYTHING TO THE FIRST NIX REGIONS.
      IF(NIX) 521,521,518
  518 DO 520 I=1,NIX
      L2=L2+NZONES(I)
  520 CONTINUE
      LA=L2+1
      LB=L2+2
      ISTART=NIX+1
C
  521 DO 550 I=ISTART,IMAX
      NZHALF=NZONES(I)/2
      IF(NZONES(I)-2*NZHALF) 522,522,531
C
C       PQ AND DUDT WILL BE RECALCULATED ANYWAY IN MOTION
C       AND ARE DONE HERE ONLY TO MAKE THE NEXT PRINTOUT CONSISTENT.
C
  522 IF(I-1) 599,523,524
  523 L1=1
      L2=NZHALF
      LA=1
      LB=2
      GO TO 525
```

```
                                                                        REZ2
  524 L1=L2+1
      L2=L2+NZHALF
      LA=LA+2
      LB=LA+1
  525 NZONES(I)=NZHALF
C       COMBINE ZONES LA AND LB TO FORM ZONE L.
      X(L1)=X(LA)
      U(L1)=U(LA)
      DUDT(L1)=DUDT(LA)
      DO 530 L=L1,L2
      DMT=GRAMS(LA)+GRAMS(LB)
      X(L+1)=X(LA+2)
      U(L+1)=U(LA+2)
      DUDT(L1)=DUDT(LA+2)
      J=L+1
      NGEOM=2
          CALL GEOM
      D(L)=DMT/VOL
      E(L)=(E(LA)*GRAMS(LA) +E(LB)*GRAMS(LB))/DMT
      P(L)=(P(LA)*GRAMS(LA) +P(LB)*GRAMS(LB))/DMT
      TEMP(L)=-0.
      DTZJ(L)=-0.
      Q(L)=(Q(LA)+Q(LB))*.5
      GRAMS(L)=DMT
      PQ(L)=P(L)+Q(L)
      GAMMAJ(L)=(GAMMAJ(LA)+GAMMAJ(LB))*.5
      IF(L-L2) 526,530,530
  526 LA=LA+2
      LB=LA+1
  530 CONTINUE
      GO TO 550
C
C     COMBINE THE FIRST THREE ZONES OF A REGION HAVING AN ODD NUMBER
C     OF ZONES.
C
  531 IF(I-1) 599,535,537
  535 IF(NZONES(1)-1) 599,548,536
  536 L1=1
      L2=1
      LA=1
      LB=2
      LC=3
      GO TO 538
  537 IF(NZONES(I)-1) 599,5380,5370
 5380 L1=L2+1
      L2=L2+1
      LA=LA+1
      LB=LB+1
      GO TO 550
 5370 L1=L2+1
```

```
      L2=L1                                                              REZ2
      LA=LB+1
      LB=LA+1
      LC=LA+2
  538 DMT=GRAMS(LA)+GRAMS(LB)+GRAMS(LC)
      X(L1+1)=X(LA+3)
      U(L1+1)=U(LA+3)
      J=L1+1
      NGEOM=2
          CALL GEOM
      D(L1)=DMT/VOL
      U(L1)=U(LA)
      DUDT(L1)=DUDT(LA)
      Q(L1)=(Q(LA)+Q(LB)+Q(LC))/3.
      E(L1)=(E(LA)*GRAMS(LA) +E(LB)*GRAMS(LB) +E(LC)*GRAMS(LC))/DMT
      P(L1)=(P(LA)*GRAMS(LA) +P(LB)*GRAMS(LB) +P(LC)*GRAMS(LC))/DMT
      TEMP(L1)=-0.
      DTZJ(L1)=-0.
      GRAMS(L1)=DMT
      PQ(L1)=P(L1)+Q(L1)
      GAMMAJ(L1)=(GAMMAJ(LA)+GAMMAJ(LB)+GAMMAJ(LC))/3.
      NZONES(I)=NZHALF
      IF(NZONES(I)-1) 541,541,540
  540 L1=L1+1
      L2=L1+NZHALF-2
      LA=LC+1
      LB=LC+2
      GO TO 525
C
  541 L2=L1
      LA=LC-1
      GO TO 550
C
  548 L1=1
      L2=1
      LA=0
      LB=1
  550 CONTINUE
C
C       CREATE A NEW REGION (IMAX+1).  NEW REGION BEGINS WITH ZONE LNEW.
C       ITS INNER BOUNDARY IS INTERFACE LNEW.
C
       LNEW=L2+1
      NZONES(IMAX+1)=JLAST-LNEW
      XCONST=X(LNEW-1)-X(LNEW-2)
C       USE ARDC AMBIENT CONDITIONS FOR NEW ZONES IF ARDC IS USED.
      JLAST1=JLAST-1
      X(LNEW)=X(LNEW-1)+XCONST
      IF(IARDC.GE.31) GO TO 561
  555 DO 560 L=LNEW,JLAST1
```

```
      X(L+1)=X(L)+XCONST                                                REZ2
      ALTCM=COSPHI*(X(L)+X(L+1)) +HOBKM*1.E+5
          CALL ARDC(ALTCM,PRATIO,TRATIO,DRATIO)
      D(L)=DRATIO*.001225
      E(L)=PRATIO*1.01325E6/D(L)/.4
  560 CONTINUE
      GO TO 563
C
C       ZONE JLAST1 STILL CONTAINS AMBIENT DATA FROM BEFORE REZONING.
  561 DO 562 L=LNEW,JLAST1
      D(L)=D(JLAST1)
      E(L)=E(JLAST1)
      X(L)=X(L-1)+XCONST
  562 CONTINUE
  563 DO 564 L=LNEW,JLAST1
      PQ(L)=PQ(JLAST1)
      TEMP(L)=-0.
      GAMMAJ(L)=-0.
      DTZJ(L)=-0.
      Q(L)=0.
      U(L)=U(JLAST)
  564 CONTINUE
      X(JLAST)=X(JLAST1)+XCONST
C
  571 DO 572 L=LNEW,JLAST1
      NGEOM=2
      J=L+1
          CALL GEOM
      GRAMS(L)=D(L)*VOL
  572 CONTINUE
      NEQST(IMAX+1)=NEQST(IMAX)
      CINQ(IMAX+1)=CINQ(IMAX)
      AINQ(IMAX+1)=AINQ(IMAX)
      GAMMAI(IMAX+1)=GAMMAI(IMAX)
      IMAX=IMAX+1
      OUTBDY(IMAX)=X(JLAST)
      JUTEST=(JUTEST-LZONES)/2+LZONES
      DO 573 I=1,IMAX
  573 INTERJ(I+1)=INTERJ(I)+NZONES(I)
      TLIST(1)=2.5*TLIST(1)
      TLIST(3)=2.5*TLIST(3)
      TLIST(5)=2.5*TLIST(5)
      TLIST(6)=2.5*TLIST(6)
      NPRINT=1
      MAS=1
          CALL OUT1
  599 RETURN
        END
```

```
$IBFTC REZ3    LIST,REF
      SUBROUTINE REZ3                                                   REZ3
C       THIS SUBROUTINE MAINTAINS FINE ZONES IN THE SHOCK FRONT.
C     *****CAUTION***** REZ3 CANNOT BE USED IN SAME PROBLEM WITH REZ2.
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
      EQUIVALENCE (NELL,KREZ3),(NANCY,KREZ3A),(NOLA,KREZ3B)
C
C       NELL = NUMBER OF FINE ZONES PER LARGE ZONE.
C       NANCY =TOTAL NUMBER OF FINE ZONES.
C       NOLA  =OUTER INTERFACE OF OUTERMOST FINE ZONE.
C       NOREEN= NUMBER OF ZONES BACK FROM NOLA FOR MAKING REZONE TEST.
C
      IF(IDENT.NE.1) GO TO 603
  601 WRITE(6,602)                                                      *
  602 FORMAT(55H0SUBROUTINE REZ3 VERSION  3,  3- 1-65  TAPES 6, 26     )
      MZOAN=0
      NOREEN=3
      RETURN
C
  603 NORA=NOLA-NOREEN
C       TRIGGER REZONING IF ZONE OF MAX Q IS AT OR BEYOND ZONE NORA.
C       FIND WHERE QMAX IS.
      JMAXQ=1
      QMAX=0.
      DO 605 J=1,JCALC
      IF(Q(J).LE.QMAX) GO TO 605
  604 QMAX=Q(J)
      JMAXQ=J
  605 CONTINUE
      IF(JMAXQ.GE.NORA) GO TO 607
C       IF NC(8)=0, TRIGGER REZONING IF PQ(NORA) EXCEEDS PQ(NORA+1)
C       BY 20 PERCENT.
      IF(NC(8).NE.0) RETURN
      IF(PQ(NORA).LT.1.2*PQ(NORA+1)) RETURN
C
  607 JSTART=NOLA-NANCY
      JEND=JSTART+NELL-1
C       PRINT BEFORE AND AFTER EACH REZONING IF NC(16)=1.
      IF(NC(16).NE.1) GO TO 611
  608 NPRINT=1
      MAS=1
          CALL OUT1
C
C       FORM ONE LARGE ZONE FROM FIRST NELL FINE ZONES
  611 ZNELL=NELL
      PTOT=0.
      GTOT=0.
      TOTIE=0.
```

```
      TOTKE=0.                                                        REZ3
  615 DO 620 J=JSTART,JEND
      PTOT=PTOT+P(J)
      GTOT=GTOT+GRAMS(J)
      USQ=.5*(U(J)**2+U(J+1)**2)
      TOTIE=TOTIE+E(J)*GRAMS(J)
      TOTKE=TOTKE+.5*USQ*GRAMS(J)
  620 CONTINUE
      TOTE=TOTIE+TOTKE
      GRAMS(JSTART)=GTOT
      X(JSTART+1)=X(JEND+1)
      J=JSTART+1
      NGEOM=2
          CALL GEOM
      D(JSTART)=GRAMS(JSTART)/VOL
      P(JSTART)=PTOT/ZNELL
      Q(JSTART)=0.
      PQ(JSTART)=P(JSTART)
C       ADJUST INTERNAL ENERGY SO AS TO CONSERVE TOTAL ENERGY.
      JPNELL=JSTART+NELL
      EKJST=    (U(JSTART)**2+U(JPNELL)**2)*GRAMS(JSTART)/4.
      EIJST=TOTE-EKJST
      E(JSTART)=EIJST/GRAMS(JSTART)
C
C       RELABEL EXISTING FINE ZONES
      JENDP=JEND+1
      NOLA1=NOLA-1
  634 DO 635 J=JENDP,NOLA1
      JNEW=J-NELL+1
      X(JNEW)=X(J)
      U(JNEW)=U(J)
      D(JNEW)=D(J)
      P(JNEW)=P(J)
      Q(JNEW)=Q(J)
      E(JNEW)=E(J)
      GRAMS(JNEW)=GRAMS(J)
      PQ(JNEW)=PQ(J)
  635 CONTINUE
      JNEW=NOLA-NELL+1
      X(JNEW)=X(NOLA)
      U(JNEW)=U(NOLA)
C
C       DIVIDE NEXT LARGE ZONE INTO NELL FINE ZONES.
C       NOLA STILL REFERS TO THE LARGE ZONE ABOUT TO BE DIVIDED.
  640 JDONE=NOLA+1
      UAMB=U(NOLA+1)
      DAMB=D(NOLA)
      PAMB=P(NOLA)
      EAMB=E(NOLA)
      XSIZE=(X(NOLA+1)-X(NOLA))/ZNELL
```

```
C       ADJUST INTERPOLATION IF NOLA+1 IS LAST INTERFACE IN PROBLEM.    REZ3
      IF(NOLA.GE.JLAST-1) GO TO 642
  641 NYMPH=NOLA+1
      GO TO 643
  642 NYMPH=NOLA- 1
  643 X1A=(X(NOLA)+X(NOLA+1))/2.
      X1B=(X(NYMPH)+X(NYMPH+1))/2.
      X1D=X1B-X1A
C
      JBEGIN=NOLA-NELL+2
      DO 660 J=JBEGIN,JDONE
      J=J
      JMH=J-1
      X(J)=X(J-1)+XSIZE
      U(J)=UAMB
      IF(IARDC.LT.31) GO TO 6431
 6430 P(JMH)=PAMB
      D(JMH)=DAMB
      E(JMH)=EAMB
      GO TO 6450
C       INTERPOLATE WITHIN THE LARGE ZONE IF ARDC IS USED.
 6431 DINT=D(NYMPH)
      PINT=P(NYMPH)
      EINT=E(NYMPH)
      X1C=X(J)-XSIZE/2.
      X1R=(X1C-X1A)/X1D
  645 D(JMH)=DAMB+(DINT-DAMB)*X1R
      P(JMH)=PAMB+(PINT-PAMB)*X1R
      E(JMH)=EAMB+(EINT-EAMB)*X1R
 6450 Q(JMH)=0.
      PQ(JMH)=P(JMH)
      PSI(JMH)=PQ(JMH)*14.50382E-6
      TEMP(JMH)=0.
      DTZJ(JMH)=DTZJ(NOLA)
      GAMMAJ(JMH)=GAMMAJ(NOLA)
C
  646 NGEOM=2
          CALL GEOM
      GRAMS(JMH)=D(JMH)*VOL
  660 CONTINUE
C
C       ADJUST NZONES AND INTERJ.
 1000 NELL1=NELL-1
C       NLAST IS INNER BOUNDARY OF FINE ZONES.
      NLAST=NOLA-NANCY
      DO 1020 II=2,IMAX
      IF(NOLA.LT.INTERJ(II)) GO TO 1020
      IF(NLAST.GE.INTERJ(II)) GO TO 1020
 1002 INTERJ(II)=INTERJ(II)-NELL1
      NZONES(II)=NZONES(II)+NELL1
```

```
      NZONES(II-1)=NZONES(II-1)-NELL1                                   REZ3
 1020 CONTINUE
C
  661 NOLA=NOLA+1
      MZOAN=MZOAN+1
      IF(NC(14).EQ.0) GO TO 665
      WRITE(26,666)  MZOAN,NOLA,NCYCLE                                 *
      GO TO 6660
  665 WRITE(6,666)   MZOAN,NOLA,NCY(                                   *
  666 FORMAT(25HOREZONING BY REZ3, MZ..N=I3,6H NOLA=I3,8H NCYCLE=I4)
 6660 IF(NOLA-JLAST) 669,667,667
  667 KREZ3=0
      IF(NC(14).EQ.0) GO TO 6670
      WRITE(26,668)                                                     *
C       WRITE TAPE 6 ALSO.
 6670 WRITE(6,668)                                                      *
  668 FORMAT(36H1FINE ZONES HAVE REACHED OUTER WALL.  /53HO KREZ3 IS NOW
     1 ZERO TO PREVENT FURTHER USE OF REZ3.  )
  669 IF(NC(16).NE.1) RETURN
  670 NPRINT=1
      MAS=1
          CALL OUT1
  699 RETURN
       END
```

```
$IBFTC REZ4    LIST,REF
      SUBROUTINE REZ4                                                   REZ4
C                                                  REZONE FIRST REGION
C
C     THIS SUBROUTINE REDUCES THE NUMBER OF ZONES IN REGION 1 TO KREZ4
C     ZONES AT TIME TREZO. REGION 1 OFTEN KEEPS THE TIMESTEP SMALL AFTER
C     ALL PHENOMENA OF INTEREST HAVE MOVED OUT INTO OTHER REGIONS.  THIS
C     ROUTINE PERMITS THE PROBLEM TO RUN FASTER IN SUCH CASES.
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
      IF(IDENT.NE.1) GO TO 403
  401 WRITE(6,402)                                                      *
  402 FORMAT(40HOSUBROUTINE REZ4 VERSION  3,  3- 1-65    )
      NDONE=0
      RETURN
C
C        DO NOT PROCEED UNTIL T HAS REACHED TREZO.
  403 IF(T.LT.TREZO) RETURN
C        DO NOT PROCEED UNLESS PROBLEM HAS SPHERICAL SYMMETRY.
      IF(K.NE.3) RETURN
C        DO NOT PROCEED IF KREZ4 IS NOT LESS THAN NZONES(1).
      IF(NZONES(1).LE.KREZ4) RETURN
  405 EKIN1=0.
      PTOT=0.
      GTOT=0.
      TOTIE=0.
      TOTKE=0.
      JSTOP=NZONES(1)
      ZN=NZONES(1)
      NPRINT=1
      MAS=1
          CALL OUT1
      WRITE(6,415)   JSTOP,KREZ4,NCYCLE                                 *
  415 FORMAT(37HOREGION 1 IS ABOUT TO BE REDUCED FROM,I4,3H TO,I4,15H ZO
     1NES AT CYCLE,I5)
      DO 420 J=1,JSTOP
      PTOT=PTOT+P(J)
      GTOT=GTOT+GRAMS(J)
      USQ=.5*(U(J)**2+U(J+1)**2)
      TOTIE=TOTIE+E(J)*GRAMS(J)
      TOTKE=TOTKE+.5*USQ*GRAMS(J)
  420 CONTINUE
C
      ETOT1=TOTIE+TOTKE
      III=INTERJ(2)
      XXX=X(III)
      ZZZ=KREZ4
      SIZE=XXX/ZZZ
      VOL1=4.1887902*XXX**3
```

```
      DAV=GTOT/VOL1                                                     REZ4
      PAV=PTOT/ZN
      DO 430 J=1,KREZ4
      X(J+1)=X(J)+SIZE
      VOLUME=4.1887902*(X(J+1)**3-X(J)**3)
      GRAMS(J)=DAV*VOLUME
      P(J)=PAV
      Q(J)=0.
      U(J)=U(JSTOP)*X(J)/XXX
      D(J)=DAV
      DTZJ(J)=-0.
      TEMP(J)=-0.
      PQ(J)=P(J)
      EKIN1=EKIN1+(U(J)**2+U(J+1)**2)*GRAMS(J)/4.
  430 CONTINUE
C
C       ADJUST E TO CONSERVE TOTAL ENERGY IN REGION 1 .
      EINT1=ETOT1-EKIN1
      EPERG=EINT1/GTOT
      DO 440 J=1,KREZ4
  440 E(J)=EPERG
C
C       RELABEL REST OF ZONES
      JSTART=KREZ4+1
      JC=JSTOP-KREZ4
      JEND=JLAST-JC
      NZONES(1)=KREZ4
      DO 450 J=JSTART,JEND
      JW=J+JC
      X(J)=X(JW)
      U(J)=U(JW)
      D(J)=D(JW)
      E(J)=E(JW)
      Q(J)=Q(JW)
      PQ(J)=PQ(JW)
      GRAMS(J)=GRAMS(JW)
      GAMMAJ(J)=GAMMAJ(JW)
      DTZJ(J)=DTZJ(JW)
      DUDT(J)=DUDT(JW)
      DWAS(J)=DWAS(JW)
      EWAS(J)=EWAS(JW)
      PWAS(J)=PWAS(JW)
      TEMP(J)=TEMP(JW)
  450 CONTINUE
C
      IMAX1=IMAX+1
      DO 460 I=2,IMAX1
  460 INTERJ(I)=INTERJ(I)-JC
      JLAST=JLAST-JC
      JLAST1=JLAST-1
```

```
      JCALC=JCALC-JC
      KREZ3B=KREZ3B-JC
      NPRINT=1
      MAS=1
          CALL OUT1
C       PREVENT FURTHER USE OF THIS ROUTINE BY SETTING KREZ4=0.
      KREZ4=0
  499 RETURN
       END
```

```
$IBFTC ARDC    LIST,REF
      SUBROUTINE ARDC(ALTCM,PRATIO,TRATIO,DRATIO)                        ARDC
C     REFERENCE
C          MINZNER, R.A. AND RIPLEY, W.S., THE ARDC MODEL ATMOSPHERE,
C          1956, AFCRC TN-56-204, AD-110233, DEC 56, UNCLASSIFIED.
C     ALTCM=ALTITUDE IN CENTIMETERS
C     PRATIO=RATIO OF AMBIENT PRESSURE TO SEA LEVEL PRESSURE (1 ATM).
C     TRATIO=RATIO OF AMBIENT TEMPERATURE TO SEA LEVEL TEMP (288.16 K).
C     DRATIO=RATIO OF AMBIENT DENSITY TO SEA LEVEL DENSITY(.001225 G/CC)
C
  100 ALTZ=ALTCM/100.
      ALTH=6356766.0*ALTZ/(6356766.0+ALTZ)
      IF(ALTH.GT.11000.) GO TO 102
  101 TEMP=288.16-0.0065*ALTH
      PAMB=14.696178/(288.160/(288.160-0.0065*ALTH))**5.25612218
      GO TO 118
  102 IF(ALTH.GT.25000.) GO TO 104
  103 TEMP=216.66
      PAMB=3.28254528/(10.**(0.068483253E-3*(ALTH-11000.0)))
      GO TO 118
  104 IF(ALTH.GT.47000.) GO TO 106
  105 TEMP=216.66+0.003*(ALTH-25000.0)
      PAMB=0.36094654/((141.660+3.0E-3*ALTH)/216.66)**11.38826473
      GO TO 118
  106 IF(ALTH.GT.53000.) GO TO 108
  107 TEMP=282.66
      PAMB=0.0174686/(10.**(0.0524926823E-3*(ALTH-47000.0)))
      GO TO 118
  108 IF(ALTH.GT.79000.) GO TO 110
  109 TEMP=282.66-0.0045*(ALTH-53000.0)
      PAMB=8.40408E-3/((282.66/TEMP)**7.592176)
      GO TO 118
  110 IF(ALTH.GT.90000.) GO TO 112
  111 TEMP=165.66
      PAMB=1.46198E-4*EXP (-0.0341647942*(ALTH-79000.0)/165.66)
      GO TO 118
  112 IF(ALTH.GT.105000.) GO TO 114
  113 TEMP=165.66+0.0040*(ALTH-90000.0)
      PAMB=1.5519E-5*(165.66/TEMP)**8.541198
      GO TO 118
  114 IF(ALTH.GT.160000.) GO TO 116
  115 TEMP=225.66+0.02*(ALTH-105000.0)
      PAMB=1.04442E-6*(225.66/TEMP)**1.708239
      GO TO 118
  116 IF(ALTH.GT.170000.) GO TO 119
  117 TEMP=1325.66+0.01*(ALTH-160000.0)
      PAMB=5.14015E-8*(1325.66/TEMP)**3.4164794
      GO TO 118
  119 IF(ALTH.GT.200000.) GO TO 121
  120 TEMP=1425.66+0.005*(ALTH-170000.0)
```

```
      PAMB=4.0654E-8*(1425.66/TEMP)**6.832958
      GO TO 118
  121 TEMP=1575.66+0.0035*(ALTH-200000.0)
      PAMB=2.0595E-8*(1575.66/TEMP)**9.761369
  118 TRATIO=TEMP/288.16
      PRATIO=PAMB/14.696178
      DENSAM=3.2365427E-4*PAMB/TEMP
C     DENSAM IS IN UNITS OF SLUGS PER SQUARE INCH PER FOOT.
      DRATIO=DENSAM/.0000165
      RETURN
      END
```

```
$IBFTC GEOM    LIST,REF
      SUBROUTINE GEOM                                                    GEOM
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
      IF(IDENT.NE.1) GO TO 9
    5 WRITE(6,6)                                                         *
    6 FORMAT(40H0SUBROUTINE GEOM VERSION  3,  3- 1-65    )
      RETURN
C
C        CALC AREA OF INTERFACE J OR VOLUME OF ZONE J-1
    9 NGEOM=NGEOM
      GO TO (10,20),NGEOM
   10 K=K
      GO TO (11,12,13),K
   11 AREA=1.
      GO TO 99
   12 AREA=6.2831853*X(J)
      GO TO 99
   13 AREA=12.566372*X(J)**2
      GO TO 99
   20 K=K
      GO TO (21,22,23),K
   21 VOL=X(J)-X(J-1)
      GO TO 99
   22 VOL=3.1415927*(X(J)**2-X(J-1)**2)
      GO TO 99
   23 VOL=4.1887902*(X(J)**3-X(J-1)**3)
   99 RETURN
      END
```

```
$IBFTC EQST    LIST,REF
      SUBROUTINE EQST                                                   EQST
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
C       AN EQUATION OF STATE MUST CALCULATE P(J), E(J), AND SOUND(J)
C       AND, IF DESIRED, TEMP(J) AND GAMMAJ(J).
C
      NQST=NQST
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12),NQST
    1     CALL EQS1
       GO TO 100
    2     CALL EQS2
       GO TO 100
    3     CALL EQS3
       GO TO 100
    4     CALL EQS4
       GO TO 100
    5     CALL EQS5
       GO TO 100
    6     CALL EQS6
      GO TO 100
    7     CALL EQS7
      GO TO 100
    8     CALL EQS8
      GO TO 100
    9     CALL EQS9
      GO TO 100
   10     CALL EQ10
      GO TO 100
   11     CALL EQ11
      GO TO 100
   12     CALL EQ12
  100 RETURN
      END
```

```
$IBFTC EQS1    LIST,REF
      SUBROUTINE EQS1                                                   EQS1
C
C     EQUATION OF STATE FOR GAMMA-LAW GAS, E=PV/(G-1).
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
C     AN EQUATION OF STATE MUST CALCULATE P, E, AND SOUND.
C     IF DESIRED, IT MAY ALSO CALCULATE TEMP AND GAMMAJ.
C
      IF(IDENT.NE.1) GO TO 800
      IF(NOMORE.EQ.71) RETURN
  101 WRITE(6,102)                                                      *
  102 FORMAT(40H0SUBROUTINE EQS1 VERSION  3,  3- 1-65    )
      RGAS=2.87043E+6
      NOMORE=71
  103 RETURN
C
  800 GAMMA=GAMMAI(I)
C       BEGIN ENERGY ITERATION.
   90 GA=GAMMA-1.0
      GB=(GAMMA+1.)/GA
      V1=1./D(JMH)
      V2=1./DWAS(JMH)
      VB=V1-V2
      VC=GB*V1-V2
      P(JMH)=(2.*EWAS(JMH) -(PWAS(JMH)+2.*Q(JMH))*VB)/VC
      PV=P(JMH)/D(JMH)
      E(JMH)=PV/GA
      TEMP(JMH)=PV/RGAS
      GPV=GAMMA*PV
      IF(GPV) 211,211,220
  211 WRITE(6,212)  JMH,E(JMH),D(JMH),P(JMH),GAMMA
  212 FORMAT(18H1*****STOP IN EQS1/6H ZONE=I4/8H E(JMH)=1PE14.5/8H D(JMH
     1)=1PE14.5/8H P(JMH)=1PE14.5/8H GAMMA =1PE14.5)
      KUTOFF=1
      MAS=1
          CALL OUT1
  220 SOUND(JMH)=SQRT(GPV)
      GAMMAJ(JMH)=GAMMA
      RETURN
      END
```

```
$IBFTC EQS2    LIST,REF
      SUBROUTINE EQS2                                                   EQS2
C
C     APPROXIMATE EQUATION OF STATE FOR REAL AIR (VARIABLE GAMMA).
C
*****   INSERT STANDARD COMMON CARDS BEFORE COMPILING   *****
C
      DATA RGAS/2.87043E+6/
C
      IF(IDENT.NE.1) GO TO 800
      IF(NOMORE.EQ.73) RETURN
  101 WRITE(6,102)                                                      *
  102 FORMAT(45HOSUBROUTINE EQS2 VERSION  3,  3- 1-65  AIR   )
      NOMORE=73
  103 RETURN
C
C       CHANGE ENERGY FROM ERG/GRAM TO MBAR CC/GRAM.
  800 ENERGY=E(JMH)*1.E-12
      DENRAT=D(JMH)/.001225
      IF(ENERGY.GT.0.) GO TO 4
  202 WRITE(6,212)  JMH,E(JMH),D(JMH),P(JMH),GAMMA
  212 FORMAT(18H1*****STOP IN EQS2/6H ZONE=I4/8H E(JMH)=1PE14.5/8H D(JMH
     1)=1PE14.5/8H P(JMH)=1PE14.5/8H GAMMA =1PE14.5)
      KUTOFF=1
      MAS=1
          CALL OUT1
      STOP
    4 IF(ENERGY.GT.0.003) GO TO 204
    2 GAMMA=1.4
    3 GO TO 90
  204 ELOG=ALOG10(ENERGY)
      IF(DENRAT) 202,202,205
  205 DLOG=ALOG10(DENRAT)
    6 IF(ENERGY.GT.0.015) GO TO 8
    7 GAMMA=-.071533*ELOG+1.21953
      GO TO 90
    8 IF(ENERGY.GT.0.05) GO TO 10
    9 AG=-.172124
      BG=1.03606
      CG=-.338509
      DG=.732590
      GO TO 80
   10 IF(ENERGY.GT.0.3) GO TO 12
   11 AG=-.109233
      BG=1.11789
      CG=-.106663
      DG=1.03423
      GO TO 80
   12 IF(ENERGY.GT.20.) GO TO 14
   13 AG=.0411205
```

EQS2

```
      BG=1.1965
      CG=.0137068
      DG=1.0972
      GO TO 80
   14 IF(ENERGY.GT.70.) GO TO 16
   15 AG=.27569
      BG=.89132
      CG=.52383
      DG=.43348
      GO TO 80
   16 GAMMA=.086588*ELOG+1.24024
      IF(GAMMA.GT.1.667) GAMMA=1.667
      GO TO 90
   80 GAM1=AG*ELOG+BG
      GAM2=CG*ELOG+DG
      GAMMA=GAM1-(1.-DLOG)/7.*(GAM1-GAM2)
C       CALCULATION OF AN EFFECTIVE GAMMA NOW COMPLETED.
C
C       BEGIN ENERGY ITERATION.
   90 GA=GAMMA-1.0
      GB=(GAMMA+1.)/GA
      V1=1./D(JMH)
      V2=1./DWAS(JMH)
      DV=V1-V2
      VC=GB*V1-V2
      P(JMH)=(2.*EWAS(JMH)-(PWAS(JMH)+2.*Q(JMH))*DV)/VC
C
   22 PV=P(JMH)/D(JMH)
      E(JMH)=PV/GA
      TEMP(JMH)=PV/RGAS
      GPV=GAMMA*PV
      IF(GPV) 202,202,221
  221 SOUND(JMH)=SQRT(GPV)
      GAMMAJ(JMH)=GAMMA
  999 RETURN
      END
```